

# Trabajo Fin de Grado

Criticalidad en redes neuronales: comparación  
entre modelos y datos biológicos

Criticality in neural networks: comparison  
between neural data and models

Autor

Marcos José Negre Saura

Director

Manuel González Bedia

Escuela de ingeniería y arquitectura  
2017





Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza

## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Marcos José Negre Saura,

con nº de DNI 73159104Z en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado, (Título del Trabajo)

Criticalidad en redes neuronales: comparación entre modelos y datos biológicos

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 22 de Noviembre

Fdo: Marcos José Negre Saura



## Índice

Base teórica y objetivos .....	6
Introducción .....	7
Principios básicos .....	7
Conceptos.....	8
El modelo de Ising .....	9
Ising-Monte Carlo-Metrópolis.....	12
Primera parte .....	14
Análisis de las correlaciones neuronales en función de la temperatura .....	14
Código.....	14
Resultados .....	16
Ajuste de la dependencia de las correlaciones con la distancia .....	18
Código.....	19
Resultados .....	20
Influencia de la magnitud de la red.....	22
Código.....	23
Resultados .....	24
Segunda parte .....	26
Código .....	26
Resultados .....	26
Picos de actividad e interacción .....	26
Ajuste del modelo .....	29
Coste computacional y otras dificultades .....	32
Conclusión .....	33
Bibliografía .....	36
Anexos .....	38
Código de simulación de la red neuronal.....	38
Código de análisis de entropía y magnetización .....	44
Código de análisis de datos biológicos.....	50



## Base teórica y objetivos

La motivación de este estudio radica en entender cómo se procesa la información en redes neuronales, y cómo se distribuye la actividad espacialmente. Además, se estudiarán datos de neuronas biológicas a través de modelos para catalogarlos y comparar la actividad de neuronas reales con la proporcionada por los modelos de neuronas artificiales.

A lo largo de las siguientes líneas se hablará sobre el análisis de redes neuronales con diferentes modelos. Todos ellos giran en torno a los mismos principios, que son los mecanismos de aprendizaje basados en la Máquina de Boltzmann o modelo de Ising. Estos modelos son sistemas de máxima entropía. Su relevancia e interés en la actualidad a la hora de construir redes neuronales se debe a su rapidez frente a otros modelos a la hora de simular la red, característica que se estudiará posteriormente. Dentro de los modelos de máxima entropía, se usará un modelo de Ising. La diferencia entre ambos reside en que el primero tiene solución analítica bajo ciertas condiciones, mientras que el segundo no.

Se usan modelos de máxima entropía debido a que este tipo de sistemas basados en termodinámica permiten trasladar conceptos de este campo al análisis de redes. Dentro de dicho modelo, de forma análoga a los formalismos empleados en termodinámica, utilizaremos recurrentemente la "temperatura" operativa del sistema como parámetro de control de la actividad del sistema. Es importante porque ayuda a situar y entender el comportamiento de la red neuronal conforme se modifica esta variable. Al variar este parámetro, se logrará identificar cual es el rango en el cual la red tiene un intercambio de información más eficiente, y que además se asimila al comportamiento de un conjunto de neuronas biológicas reales. La zona de variación donde aumenta la eficiencia de la red es la transición de fase, similar a lo que ocurre en materiales ferromagnéticos como se mencionará a continuación. Este punto se denomina punto crítico y se localiza entorno a la temperatura crítica. En el caso de estudio que nos ocupa se analizará la zona de criticalidad en modelos teóricos y en datos reales.

El contexto de estas investigaciones se ubica en torno a la mecánica estadística, más concretamente en el punto de criticalidad. Dicho punto es en cual se localiza la temperatura crítica. Los tejidos neuronales funcionan según diferentes estudios como una combinación de dos fases (H.Dale Beggs 2008). En resumen, estas dos fases son un sistema ordenado y un sistema completamente entrópico, desordenado. La combinación de éstas maximiza y optimiza la canalización de información entre neuronas y tejidos nerviosos, haciendo dicha transmisión de datos más eficiente. Esta localización peculiar es la que se denomina punto de criticalidad o, como ya se ha nombrado, temperatura crítica.

En la primera parte de la investigación se usará un algoritmo de tipo Monte Carlo. Es un algoritmo que se empleará para actualizar el estado del modelo de Ising, que es el tipo de red que usamos. Con dicha red se ensayará y demostrará la relación entre distancia y actividad de pares neuronales. La razón de estudio de la interacción entre ambas magnitudes es que la distribución de los resultados varía dependiendo de la zona de estudio, se entiende por zona la temperatura usada. Habrá dos leyes diferentes según la temperatura de trabajo, ley de potencias o exponencial. Todo ello se ensayará con diferentes dimensiones de redes, cantidad de actividad y distancias para arrojar luz sobre conceptos que se irán explicando y comprobando a lo largo de la investigación. Estos conceptos son: interacción y distancia entre neuronas, curvas de ajuste neuronal exponencial y potencial, entropía, capacidad calorífica, magnetización, susceptibilidad, picos de actividad, etc. Además, se usará y modificará la temperatura entorno a la temperatura crítica para valorar los resultados.

La segunda parte del estudio será una comparación con capturas de redes neuronales vivas y de los modelos ya vistos anteriormente. Esta parte tiene como misión poner en contraste datos biológicos junto con los experimentos de neuronas artificiales. Tras esto se llegará a una conclusión de hasta qué punto el modelo se adecua a la interacción neuronal de un ser con red neuronal compleja o cerebro.

## Introducción

### Principios básicos

En el caso que nos ocupa, queremos entender datos biológicos a través de modelos matemáticos. Para ello, se pasará a explicar el tipo de modelo usado para la investigación.

La Máquina de Boltzmann o modelo de Ising son sistemas computacionales encargados de resolver restricciones de carácter débil. La diferencia con otros modelos matemáticos de resolución de restricciones es que, a diferencia del sistema citado, usaban tipos de restricciones fuertes (Waltz; 1975. Winston; 1984). Estas restricciones de carácter fuerte son rígidas, es decir, deben ser satisfechas por cualquiera de las soluciones. Un ejemplo de restricción fuerte sería la construcción de un puzle en el cual se tiene que seguir un orden causal unívoco, singular, al construirlo y encajar todas las piezas. Sin embargo, un sistema de restricciones débiles podría asimilarse a la interpretación de una imagen poco definida, en la cual, por ausencia de restricciones visuales concretas, da lugar a diferentes interpretaciones y a diferentes soluciones. Habrá una solución preferible con la que concuerdan la mayoría de las restricciones, aunque siempre violara alguna de dichas restricciones débiles por no poder demostrar su similitud al cien por cien.

La máquina está compuesta por elementos computacionales individuales y diferenciados que representan las neuronas aisladas. Estas unidades se conectan entre sí por uniones bidireccionales. Como se introducirá más adelante las neuronas están conectadas unas a otras con enlaces que difieren en su peso. Éste peso es lo que mide lo fuerte que es la conexión entre dos neuronas y por ello como de relaciones están sus respuestas a los estímulos. Las unidades tienen dos posibles estados, *on* y *off*, dichos estados son un punto clave de los modelos que se explicarán para la construcción de los programas de aprendizaje y análisis de las redes neuronales. Éstos posibles estados pueden tomar valores reales o signos. En nuestro objeto de estudio primero se verá que en los modelos se tratan como direcciones de alineación de unas unidades llamadas *spins* para pasar a ser usados en lenguaje neuronal como estados (+1, -1) de activación. También puede trasladarse estos valores a estados *on* y *off* que se refieren a la aceptación o no de una hipótesis, es decir, si dos unidades están en modo *on* aumenta la posibilidad de que ambas acepten dicha hipótesis, mientras que si cualquiera de ellas está en modo *off* es bastante más probable que una de ellas acepte dicha hipótesis y la otra la rechace. Que las respuestas a los estímulos de dos neuronas sean similares también depende del peso de la fuerza de sus uniones.

En estas líneas se habla solo de forma teórica sin mostrar ningún modelo concreto de probabilidad del que depende la aceptación o no de las hipótesis. A continuación, se pasará a profundizar en los términos y cómo funciona nuestra red de estudio. Tras esto, para el análisis de las redes neuronales se plantean modelos de probabilidad más exactos y con resultados numéricos modelables y medibles.



## Conceptos

Para hacer la lectura más sencilla antes de explicar el modelo y los algoritmos se hará una breve introducción de conceptos.

- **Modelos de máxima entropía:** Esta familia de modelos es la que engloba la teoría básica de este tipo de estudios neuronales. Hay que remarcar que se puede usar para otro tipo de aplicaciones como magnetismo y otros problemas físicos. Se basa en la mecánica estadística y establece que ante unas proposiciones y condiciones fijas la distribución de probabilidad menos oblicua o sesgada tiende a maximizar la entropía, a aumentar el desorden. En otras palabras, la solución con menos condicionantes y más simple es la que satisface mejor la situación o problema a resolver. En esto se fundamenta el **modelo de Ising**.
- **Máquina de Boltzmann:** La máquina de Boltzmann es el tipo de modelo de probabilidades usado en estos sistemas, siguiendo las características de la teoría de los **modelos de máxima entropía**. Es similar a los **modelos de Ising**. La diferencia principal es que incorpora una regla de actualización de estados, la regla de Glauber, que es similar a la regla del algoritmo de Metrópolis.
- **Modelo de Ising:** Es el modelo a usar en los ensayos y partir del cual se desarrollan diferentes algoritmos de aprendizaje o de actualización de estados como es la familia de métodos **Monte Carlo**.
- **Método de Monte Carlo:** Algoritmos de actualización de estados que usamos como herramienta para el objeto de estudio de este trabajo. Se ha usado por ejemplo para crear algoritmos como **metrópolis** de actualización neuronal que se usará de ejemplo más adelante para la explicación teórica.
- **Temperatura crítica:** La temperatura crítica es una manera de expresar el punto en el cual la actividad neuronal refleja un comportamiento óptimo. En este caso se entiende comportamiento óptimo como la maximización de características de la red, por ejemplo, la eficiencia de propagación de la información. El porqué de que este comportamiento sea deseable se debe a que en este punto conviven el estado ordenado y el entrópico o desordenado. Esto hace que la respuesta de la red se asimile a la de un sistema biológico. Puede tener otros nombres que se refieren a la misma localización, como **punto crítico** y se ubica en la transición de fase entre los dos estados ya mencionados.
- **Spin:** Habla sobre la orientación de los materiales férreos en un campo magnético. Se usa como analogía en las redes neuronales ya que en los primeros estudios que se hicieron se usaba esta terminología al hablar de mecánica estadística. Una neurona puede estar activada o no en un instante dado, igual que un electrón puede estar orientado de forma ordenada o de forma aleatoria en dichos materiales. Para una neurona se trasladará este concepto y se podrá hablar de **estado on y off, estado activo y desactivado, +1 y -1, +1 y 0**.

## El modelo de Ising

El modelo de Ising tiene sus inicios en el físico de mismo nombre, Ernst Ising, quién inventó dicho modelo para el estudio del magnetismo en relación con la estadística. Este sistema trataba de predecir la probabilidad de que la orientación de los electrones fuera la misma o aleatoria en materiales férreos. Esta interdependencia entre las orientaciones de los electrones es la manera más sencilla de entender cómo se relaciona una neurona con otra como se explicará a continuación.

El uso de las redes de Ising para el estudio neuronal se debe a que los científicos están tratando de encontrar un modelo universal con el que poder demostrar y predecir la manera en la que un colectivo neuronal se va a comportar. En otras palabras, lo que se está intentando hallar es un patrón que pueda representar cualquier sistema cerebral. Casi todos los estudios hechos hasta el momento son ensayos empíricos en los que se compara ejemplos de neuronas vivas con programas como el que se va a mostrar aquí que simulan conexiones y actividad neuronal. La razón de que la investigación siga este curso es intentar progresar en el conocimiento sobre el funcionamiento neuronal sin la necesidad de conocer los detalles microscópicos del comportamiento bioquímico y neurotransmisor de las redes neuronales biológicas ya que es bastante más complicado de analizar.

A continuación, se aborda el estudio del ferromagnetismo y de los materiales paramagnéticos. Dichos materiales tienen la capacidad de alinear sus electrones de la capa más externa en presencia de un material magnético y retornar a un estado aleatorio cuando esta fuerza desaparece. Sin embargo, para poder volver a la situación de orientación desordenada es necesario alcanzar la temperatura crítica o de *Curie* porque de otro modo el proceso no sería reversible. Es por eso que se utilizará el término de temperatura crítica como situación deseable para la maximización de la transmisividad de información en las redes neuronales. Como se ha referenciado en otros puntos, se quiere un estado intermedio entre el orden (alineación de los electrones en el modelo de Ernst Ising) y la completa aleatoriedad. La temperatura crítica tiene esta característica y permite un estado de transitoriedad.

A continuación, en la figura Imagen 1.1. se muestra la transición de fase. La magnetización representada en ordenadas, como se indica en el pie de la gráfica, alude a la orientación de los *spins*. En la fase ordenada todos se encuentran en la misma dirección. Esto se ilustra con la media de las orientaciones que da como resultado la unidad, ya que el porcentaje total de unidades igualmente orientadas es el total haciendo que la media sea la unidad. Por otra parte, en la zona desordenada la media de la orientación es nula ya que por probabilidad habrá el mismo número de *spins* en una dirección que en la contraria (+1,-1)

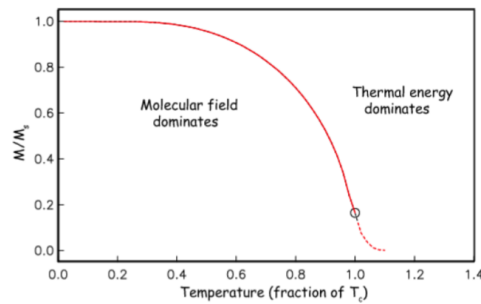


Imagen 1.1.

*Ejemplo de la distribución y de la fase de transición en la zona de temperatura crítica. En el eje de ordenadas está representada la magnetización ( $M/M_s$ ) y en el eje de abscisas la temperatura ( $\frac{T}{T_c}$ ). La magnetización cambia cuando se alcanza la transición de fase haciendo que se modifiquen los spins, pudiendo tomar una orientación cada vez más aleatoria conforme se aumenta la temperatura.*

*Imagen obtenida de Simulations: The Ising Model por Asher Preska, Steinberg, Michael Kosowsky and Seth Fraden.*

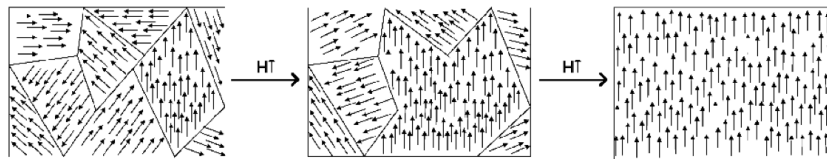


Imagen 1.2

*En estas imágenes se puede ver como la alineación de los dominios va en aumento conforme aumentamos la temperatura. El primer estado representa una situación casi totalmente aleatoria, conforme se acerca a la temperatura crítica se va viendo una situación en la que coexisten ambas formas, finalmente se alcanza un estado de ordenación total. Las líneas que circunscriben las zonas con misma orientación de los spins son las llamadas paredes de los dominios.*

*Imagen obtenida de Simulations: The Ising Model por Asher Preska, Steinberg, Michael Kosowsky and Seth Fraden.*

La transición de fase solo es visible en modelos de dos o más dimensiones, en sus inicios Wilhem Lenz y Erns Ising resolvieron el nombrado modelo en una única dimensión (lineal, una sola variable 'M') con lo que no lograron advertir ningún tipo de transición de fase. 'M' hace referencia a la magnetización. La razón de que no haya transición en el sistema de una sola dimensión es que no se le puede añadir un campo magnético externo que modifique los parámetros del sistema. En el sistema referido al aumentar la temperatura, partiendo de cero, siempre se van a crear nuevas orientaciones de los *spins* llamadas paredes de los dominios (+1, - 1). Estas paredes son las divisiones que se crean entre zonas que tienen *spins* de diferente orientación, como se muestra en la Imagen 1.2. Las paredes en el caso de sistemas de una sola

dimensión reducen la energía libre del sistema haciendo que conforme aumenta la temperatura se alcance siempre un sistema completamente aleatorio, sin dar posibilidad a más estadios posibles. La diferencia con un sistema poli dimensional es que no siempre las paredes reducen la energía interna, ya que depende de otros factores como el campo magnético.

Por ello no en todas las zonas se genera aleatoriedad desembocando en la convivencia de ambas fases como se ha nombrado al hablar de la temperatura crítica.

El paso lo dio Lars Onsager aplicando los estudios anteriores a un sistema de dos dimensiones, matricial. Este segundo descubrimiento es el usado a lo largo del proyecto para el análisis y la obtención de datos neuronales. Gracias a la Ecuación 1.1. se puede analizar la energía de la red en dos dimensiones [i,j].

$$E = -J \sum_{i,j=nn(i)}^N s_i s_j - H \sum_{i=1}^N s_i$$

Ecuación 1.1.

El primer término “sumatorio” expresa la interacción entre pares neuronales y su estado. Cada uno de ellos ( $s_i$ )- ( $s_j$ ), llamados *spin*, pueden estar en modo *Up o Down*, es decir, tomando valor 1 o -1. Hay que recalcar que el Estado + 1 significa que la neurona a la que corresponde está activada mientras que el Estado -1 hace referencia a una neurona desactivada, sin intercambio de información aparente.

La constante de interacción (J) es mayor que cero y puede tomar diferentes valores como se verá próximamente.

El segundo término de la ecuación está referido a la energía que tiene un dipolo magnético dentro de un campo donde ( $\mu$ ), que es la permeabilidad magnética, ha sido incluida dentro de (H). Éste último símbolo hace referencia al valor del campo magnético.

La ecuación anterior representa el estado de energía global. Cada una de las unidades del sistema tiene un input concreto y dicho sistema se adapta a los inputs minimizando la energía todo lo posible, dicha energía es la representada por la mencionada ecuación.

Como se ha hablado hasta ahora se está suponiendo un comportamiento de tipo binario en el cual el estado neuronal (*spins*) puede variar entre -1 y 1 adoptando solo estos valores. Una de las consecuencias de utilizar este comportamiento binario es que son fácilmente aplicables los métodos de la física estadística con los que se obtienen resultados bastante concluyentes relacionados con la capacidad de interconexión entre las neuronas y la robustez de la red en presencia de ruido. Sin embargo, debido a esta abstracción de trasladar la actividad neuronal a meros estados positivos y negativos (-1,1), resulta más complicado enlazar las previsiones de la física estadística con los modelos de neuronas biológicas reales, ya que se necesita de un proceso de adaptación de los resultados obtenidos directamente de los ensayos en redes reales al lenguaje binario en estados de las simulaciones.

### Ising-Monte Carlo-Metrópolis

Se han hecho estudios relacionados con un algoritmo llamado *Metrópolis* que utiliza redes neuronales con un método estadístico o clase de algoritmo llamado Monte Carlo. La forma clásica de obtener la probabilidad de estado de una red era:

$$P(s_i) = \frac{1}{Z} \exp \left[ \sum_{i=1}^N h_i s_i + \frac{1}{2} \sum_{i \neq j}^N J_{ij} s_i s_j \right]$$

Ecuación 1.2.

La ecuación representa los estados individuales interaccionados en un sumatorio para obtener el estado global de la red.

Donde Z es el número de iteraciones realizadas. Esta ecuación hablando ya en términos del programa y de nuestro objeto de estudio, trasciende de verse como un campo magnético y unos valores J con sentido físico. Aquí  $J_{i,j}$  describe la interacción mutua directa entre dos neuronas que ocupan dos posiciones diferentes en la red, sean cuales quiera.  $H_i$  pasa a ocupar el concepto de neurona activa o carente de actividad (*firing or silence*).

Para resolver el problema del coste computacional, aunque como se verá más adelante sigue siendo muy elevado, se han utilizado los modelos de predicción, también llamados de métodos estadísticos, de tipo Monte Carlo que se basan en la probabilidad y que se explican en los siguientes párrafos. En este tipo de algoritmo la información de la red se obtiene en la interacción de cada par de neuronas de la red de extensión N x N. Esto significa que de tratar un problema con  $2^N$  parámetros lo resumimos a  $N^2$ .

*Metrópolis* en cierta forma es una manera de simplificar el estudio de la orientación de dichas redes. El nombrado algoritmo, que ha sido estudiado en profundidad para el caso que nos ocupa, representa como converge la alineación de los *spins* a lo largo de diferentes temperaturas pasando por el estado transitorio. Esto es uno de los puntos básicos para entender cómo funciona Monte Carlo, en el cual se basa *Metrópolis*, y el porqué de su uso para el modelado de la actividad neuronal.

Centrándonos en el algoritmo de *Metrópolis*, las simulaciones que se realizan para llegar al estado final son de tipo aleatorio, parten de una orientación azarosa la cual genera una variación de energía al interaccionar con las neuronas o *spins* contiguos. Dicha variación de energía si es  $\Delta E < 0$  hace cambiar de orientación a los *spins* mientras que si por el contrario es  $\Delta E \geq 0$  la probabilidad de que cambie de orientación estará regida por la Ecuación 1.3.

$$p = e^{-\beta \Delta E}$$

Ecuación 1.3.

Tras realizar esta operación durante un número elevado de iteraciones, más de 10000, se pueden empezar obtener resultados en los experimentos. Esto se debe a que obtenemos una simulación completa suficientemente larga como para que la distribución de probabilidad ajuste la red neuronal a uno de los posibles estados, ordenado, aleatorio o transitorio.

La ecuación anterior tras realizar la simulación completa, en otras palabras, todas las iteraciones, se puede desarrollar como se ve en la Ecuación 1.2., que es la usada a lo largo de la programación. La Ecuación 1.2. muestra la distribución de probabilidades en el modelo de Ising, el algoritmo Metrópolis asegura que los estados generados en el algoritmo cumplen dicha distribución de probabilidad de Ising.

Hay que aclarar que dicho estado final depende de la temperatura elegida, si se utiliza una temperatura crítica obtendremos el estado de transición mientras que, con cualquier otra temperatura, dependiendo de cuál sea, obtendremos un estado u otro.

La amplitud de la matriz 2D usada para el estudio neuronal tiene dimensiones  $N \times N$ . Dicha matriz puede construirse en dos sentidos diferentes, periódica o con terminaciones libres. Con terminaciones libres significaría tener una matriz simple como las que se usa en álgebra o para almacenar datos en celdas por lo que las neuronas de los extremos de la malla no tienen vecinos en las localizaciones donde termina la matriz. Por tanto, para realizar un problema más real y próximo a lo que es una conexión cerebral se ha optado por usar una matriz periódica, este término significa que hay que entender la matriz como un sistema infinito interconectado entre sí donde los extremos se unen conectan unos con otros. Por poner un ejemplo, la neurona que ocupa la primera posición (1,1) estará conectada con las neuronas (1,2) y (2,1) naturalmente, pero además con (1, N) y (N, 1). Gráficamente este tipo de matriz se refiere a un sistema que adopta la forma de una esfera para lograr conectar la primera posición con la última como en el ejemplo anterior y todas las demás posiciones que carecen de los contactos en todas las direcciones por ocupar una zona límite de la red.

## Primera parte

En esta sección del trabajo se pasará a realizar la primera simulación neuronal para comparar diferentes situaciones y obtener resultados sobre la actividad neuronal variando rangos de temperatura además de otros parámetros como la distancia y tamaño de la red.

La tarea fundamental de la primera parte es analizar las correlaciones entre neuronas para detectar si el modelo está en un punto de criticalidad. La manera más eficaz de detectar si se encuentra o no en el estado de criticalidad es por medio de la comparación con las leyes potencial y exponencial. Posteriormente, los resultados obtenidos se extrapolarán a los datos neuronales reales en la segunda fase del estudio.

### Análisis de las correlaciones neuronales en función de la temperatura

El análisis de la temperatura es la parte más importante del estudio. Se trata de justificar el uso de una determinada temperatura como temperatura en el punto crítico para luego aplicar modelos matemáticos que representen la actividad neuronal en función de si se está trabajando esta temperatura o dentro del rango superior o inferior. La temperatura crítica será aquella que va a operar la red en un estado transitorio entre el estado aleatorio y el completamente ordenado estando así en la zona llamada de transición.

La distancia entre pares neuronales es el factor que influye de manera más relevante sobre la correlación e interacción de cada dupla de neuronas. La forma en que afecta la distancia a dicha interacción varía según si nos encontramos en el punto crítico o no. Es importante remarcar que cuando nos encontramos en la zona de criticalidad o transición de fase, cerca del punto crítico, la longitud de la correlación es infinita. Esto quiere decir que dos neuronas cualesquiera por muy separadas que estén van a interactuar entre sí. Sin embargo, esto no quiere decir que la interacción sea la misma entre dos neuronas muy próximas y dos que se encuentra a una distancia considerable, conforme aumenta la distancia la correlación e interacción decaen. Por el contrario, fuera de la transición de fase, la interacción decae rápidamente a cero. Dicho de otra manera, dos neuronas que se encuentre una distancia relativamente grande no tendrán ningún tipo de correlación, su interacción será nula.

Éstos dos tipos de interacciones se puede modelar con fórmulas matemáticas como se verá en el apartado de ajuste matemático. Resumiendo, el modelo potencial,  $\frac{1}{r^\eta}$ , representará la interacción en la transición de fase, donde  $r$  es la distancia entre pares de neuronas. El modelo exponencial,  $e^{-cr}$ , hará lo propio fuera de esta área. Donde la distancia ' $r$ ' viene dada por  $r = |i1 - i2| + |j1 - j2|$ , es decir la distancia en el eje vertical más la distancia en el eje horizontal.

### Código

La parte de programación consta de cuatro partes diferenciadas. Éstas partes en orden de aparición a largo del programa son: la definición del código de generación de la red y matriz neuronal  $N_{i,j}$ , la creación y acumulación de datos de la actividad en la matriz periódica de correlación de pares neuronales  $c_{ij,ij}$ , la construcción de la matriz de distancias de las parejas de neuronas de la malla periódica  $R_{ij,ij}$  y la parte final de obtención de resultados, gráficas, etc.

La primera, se trata de recorrer la matriz que hemos determinado con un número  $N^2$  de neuronas que para el caso que nos ocupa hemos decidido darle una dimensión de  $50 \times 50$ . Se realiza una actualización al azar de las neuronas que recorre la matriz una vez por cada neurona (Bucle  $N \times N$ ), logrando con esto que al aumentar el número de iteraciones todas las neuronas prácticamente se hayan actualizado el mismo número de veces por estadística.

La actualización lo que hace es obtener la actividad de la neurona a actualizar y la de las neuronas colindantes, sin olvidar que por ser matriz periódica todas van a tener las cuatro neuronas más próximas disponibles.

Tras esta parte comienza el código principal donde tenemos el bucle que actualiza la red con el número de iteraciones deseado. Se ha comprobado que se obtienen resultados suficientemente precisos a partir de 10.000 pasos. Simultáneamente, en cada una de las iteraciones se llama a la función de entrenamiento, la cual calcula los resultados que se acumulan en la matriz de interacción que muestra la activación neuronal. Dicha matriz consta de  $N^2 \times N^2$  celdas ya que trata de medir todas las posibles asociaciones neuronales. La matriz se llena haciendo el producto de la actividad de cada par neuronal de la red. Finalmente se divide por el número de pasos ya que habremos ido acumulando la actividad de cada iteración, resultando un número que estará entre cero y la unidad que significaría actividad nula o interacción máxima entre dos neuronas como se ha visto en teoría hasta ahora.

$$c_{ij} \langle S_i, S_j \rangle = \frac{1}{T} \sum_{t=1}^T S_i(t) S_j(t)$$

Ecuación 2.1.

Como aclaración  $S$  es el valor de la actividad de una neurona que puede adoptar un valor de actividad (1) o reposo (-1) y  $T$  es el número de iteraciones.

La matriz que contiene la información de distancias se basa en la suma de distancias horizontales y verticales. Para ello se crean dos matrices que contengan dichas informaciones, es decir, todas las posibles distancias entre todas las neuronas en la matriz de coordenadas que van desde la distancia unitaria hasta 50 ( $25x + 25y$ ) que son las posiciones más alejadas dentro de la red. La distancia no tiene unidades métricas, sino que una unidad equivale a un movimiento de celda ya sea en dirección horizontal o en vertical, sin existir direcciones diagonales.

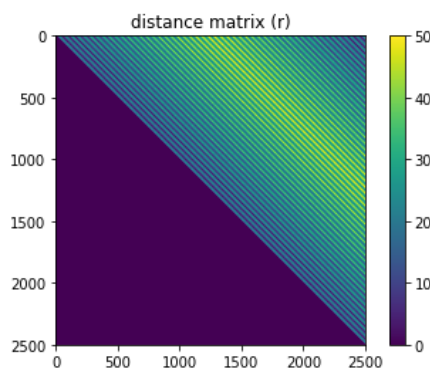


Imagen 2.1.

*En esta ilustración se representa la distancia interneuronal en una red periódica. El área que refleja una mayor separación neuronal es la ubicada en la zona intermedia de la matriz porque al ser una matriz periódica la distancia entre los límites de la matriz es cero. En otras palabras, los extremos de la red están conectados como ya se ha mencionado*



Como se muestra en la figura, la distancia mayor se da entre neuronas que se encuentren ubicadas a una distancia máxima de 25 celdas. Todas las matrices se han modificado a triangulares superiores ya que son simétricas y así se evita la duplicidad de la información.

La última parte del código contiene las imágenes de las matrices y la gráfica que aúna la distancia con la actividad. Esta gráfica es la parte más importante de este apartado porque es la que puede aportar más fácilmente información de cómo afecta la distancia a la actividad neuronal.

## Resultados

Para hacer más sencillo el análisis de los resultados se han agrupado los resultados en tres grupos diferentes. El más importante es entorno a la temperatura crítica (punto crítico) que está fijada en un valor de beta  $\beta = \frac{1}{T}$ , siendo beta el inverso de la temperatura. La temperatura crítica se puede hallar de forma teórica mediante el siguiente cálculo,  $T_c = 2/\log(1 + \sqrt[2]{2})$ . Luego se darán valores por encima del punto crítico y por debajo para ver las diferencias en el comportamiento de la red.

$$\diamond \text{ Punto crítico } \beta = \frac{1}{2,26918}$$

Como se ha explicado en teoría la situación más deseable se dará en el punto crítico. En estas gráficas se ve como la interacción es máxima en las zonas de menor distancia y va decreciendo conforme se encuentran más alejadas. Esto se ve reflejado en la matriz de distancias. Con lo cual lo que se ve es una dependencia proporcional con la distancia, pero además observando la gráfica en la que se disponen la correlación de las parejas de neuronas y su distancia relativa, se ve como a mayor distancia decrece, sin alcanzar una interacción nula en las neuronas con más alejamiento. Se puede extraer como conclusión que cualquier neurona se comunica y depende de otra por muy alejada que está en la red, con estas condiciones de análisis ( $\beta = \frac{1}{2,26918}$ ).

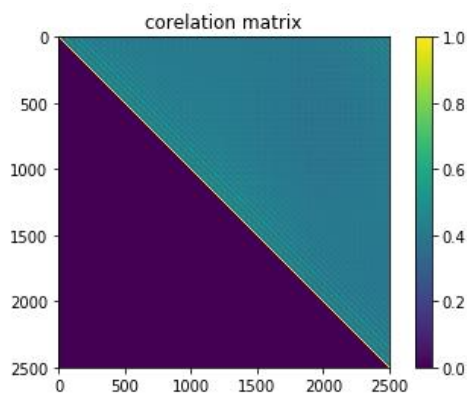


Imagen 2.2.

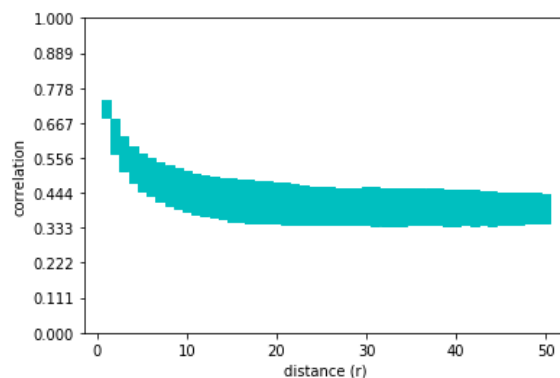


Imagen 2.3.

*Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones. Como se ve a lo largo de las ilustraciones de este análisis, cuanto mayor es la correlación en la matriz entre dos neuronas cualesquiera, más altos son los valores de correlación a pesar del aumento de la distancia interneuronal.*

- ❖ Por debajo del punto crítico  $\beta < \frac{1}{2,26918}$

Trabajar por debajo del punto crítico equivale a disminuir  $\beta$ , la razón es simplemente porque la temperatura aumenta, aumentando la entropía y sabiendo la equivalencia que tiene con beta ésta debe disminuir.

Lo que vemos aquí es como la interacción neuronal decrece muy rápido con la distancia. Al usar una temperatura muy alta lo que se consigue es hacer que las conexiones entre neuronas sean muy débiles, es decir, con mucha aleatoriedad y apenas tengan interacción con las de su entorno.

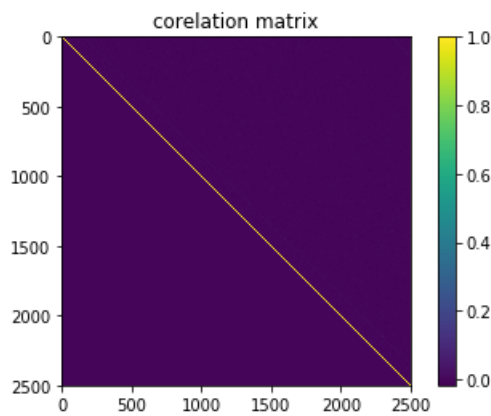


Imagen 2.4.

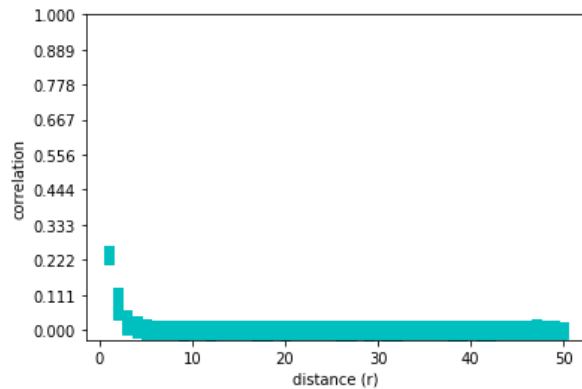


Imagen 2.5.

*Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones.  $\beta = 0,5 * \frac{1}{2,26918}$ .*

- ❖ Por encima del punto crítico  $\beta > \frac{1}{2,26918}$

En este último caso sucede el efecto contrario, disminuir la temperatura implica aumentar la capacidad de interacción y la fortaleza de las conexiones entre neuronas, haciendo la red muy rígida y provocando que las conexiones siempre estén activas, suceso tampoco deseado. Todo ello es debido al efecto contrario, la entropía disminuye haciendo que la respuesta neuronal se mantenga siempre fija.

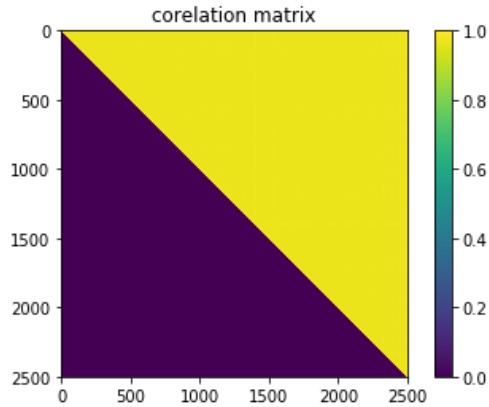


Imagen 2.6.

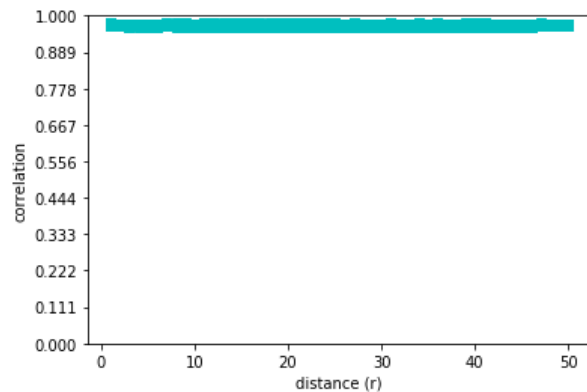


Imagen 2.7.

*Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones.  $\beta = 1,5 * \frac{1}{2,26918}$ .*

El resultado que cabe resaltar de este apartado es como varía la correlación frente a la distancia. Conforme nos aproximamos a la zonade criticalidad, es decir a la temperatura crítica, las correlaciones dejan de estar fuertemente condicionadas y empiezan a varían entre 0.8 y 0 sin llegar al valor nulo. Por fuerte condicionamiento se entienden valores de temperatura bajos que implican una red muy rígida con lo cual, siempre tiene correlaciones próximas a la unidad o valores de temperatura bajos que resultan en correlaciones prácticamente nulas. Otro aspecto a destacar es la distancia. Una de las características fundamentales del punto crítico es que la correlación, aunque desciende con la distancia, nunca alcanza el valor nulo. Este suceso significa que dos neuronas de la red intercambiarán información, aunque haya una distancia considerable entre ellas a esta temperatura. Por otro lado, en otras temperaturas o tiende a cero la correlación con la distancia o no modifica su valor debido a que la red es muy rígida.

### Ajuste de la dependencia de las correlaciones con la distancia

En este apartado se analiza como las funciones teóricas se ajustan a los ensayos realizados. Para que tenga validez el modelo, hay que asegurarse de que el punto crítico localizado actúa como tal. Para ello hoy se pueden usar herramientas como las ecuaciones que rigen el comportamiento dentro y fuera el punto crítico.

Como se ha mencionado, se utilizan dos ecuaciones que tienen validez dentro y fuera, cada una. Éstas tendrán parámetros que se obtienen de los ensayos que se llevaran a cabo, a excepción de 'r' que mide las distancias entre neuronas, ya fijadas en el análisis de la red.

La mayor diferencia entre ellas es como queda definida la interacción neuronal. Esto quiere decir que a través de estas dos ecuaciones y sus correspondientes gráficas se pueden deducir características tales como la capacidad de la red a una temperatura para comunicar información de una neurona a otra.

La ecuación que refleja el comportamiento fuera de la transición de fase donde se halla temperatura crítica, muestra un decaimiento de la correlación temprano ya que desciende de forma rápida a cero. Esto significa que conforme aumenta la distancia entre dos neuronas, la interacción decrece de forma exponencial siendo una red poco robusta. Por otra parte, dentro de la transición de fase, entre el área donde la red es más rígida y el área donde es completamente aleatoria, la pendiente es menos pronunciada. Lo más característico de esta segunda ecuación es que los valores que obtenemos de ella siempre muestran algo de interacción entre dos neuronas sean cualesquiera y halla la distancia que hay entre ellas, aunque a más distancia disminuye la correlación lógicamente.

A continuación, se definen las dos funciones de ajuste, hallando sus parámetros empíricamente, se grafican y representan junto con el error de sendas ecuaciones.

$$c_{ij} = a \bar{e}^{\gamma r} + b \quad c_{ij} = a \left( \frac{1}{r} \right)^{\gamma} + b$$

Ecuación 2.2. , Ecuación 2.3.

*A la izquierda se tiene la ecuación que define las zonas de actividad neuronal fuera del punto crítico. A la derecha la ecuación para la actividad neuronal dentro de él, en dicha temperatura. Hay que destacar que  $a$  y  $b$  son parámetros de ajuste. El parámetro ' $a$ ' debe ser lo más próximo a 1 posible, mientras que ' $b$ ' debería tender a cero.*

## Código

El código usado en este apartado se encuentra junto con el programa principal, ya que es necesario tras realizar cada ensayo ver cómo opera la red y el error cometido en el ajuste. Se trata simplemente de cribar los resultados obtenidos y realizar con ellos gráficas además de comprobaciones de ajuste.

El filtro de los resultados es quitar elementos no deseados que puedan interferir como ruido a la hora de obtener las ecuaciones de ajuste. El ruido dentro de los ensayos es normal, como se ha mencionado en otros puntos cuanto mayor número de iteraciones más definida queda la línea de datos y por tanto su modelo. Este ruido no es el que se elimina, sino que los datos eliminados son aquellos que son nulos. Por ejemplo, todos los que tengan interacción cero, como puede ser la matriz triangular inferior ya que al ser simétrica se ha decidido solo evaluar la superior y dejar la equivalente vacía con valores nulos.

## Resultados

A continuación, se muestran los resultados en los diferentes rangos de temperaturas. Las gráficas que se muestran de izquierda a derecha son: las gráficas de ajuste con los resultados de la simulación neuronal, el error cometido por cada uno de los ajustes y la matriz de interacción neuronal.

Como se puede observar en la transición de fase, el error cometido por la ecuación de ajuste en el punto crítico es sustancialmente inferior al de la curva de las zonas diferentes a la temperatura crítica. Por otro lado, en valores superiores e inferiores la gráfica exponencial proporciona un mejor ajuste.

Hay que recalcar que conforme se aproxima a la temperatura crítica el error de ambas va aumentando, aunque en comparación el de la exponencial aumente más rápidamente. Esto se debe a que el ruido aumenta y por tanto la distorsión y el error cometido también.

❖ En el punto crítico  $\beta = \frac{1}{2,26918}$

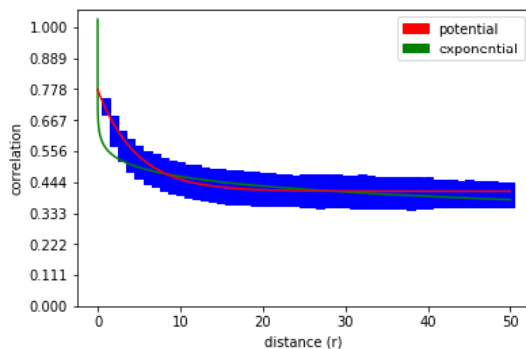


Imagen 2.8.

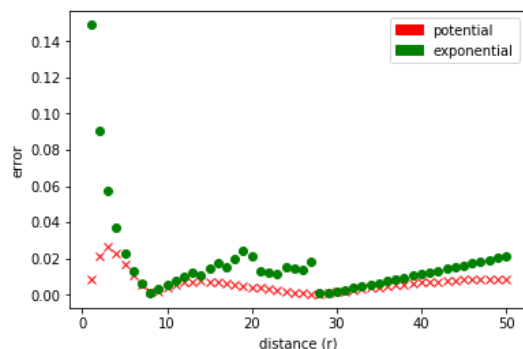


Imagen 2.9.

*Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones.  $\beta = \frac{1}{2,26918}$ .*

❖ Por debajo del punto crítico  $\beta < \frac{1}{2,26918}$

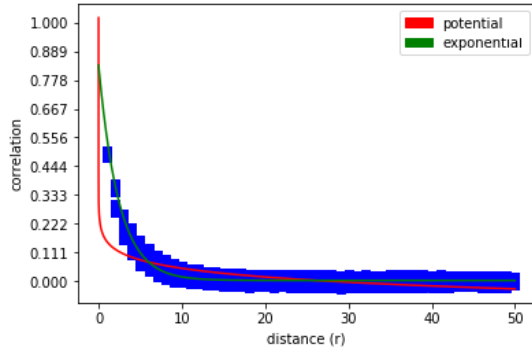


Imagen 2.10.

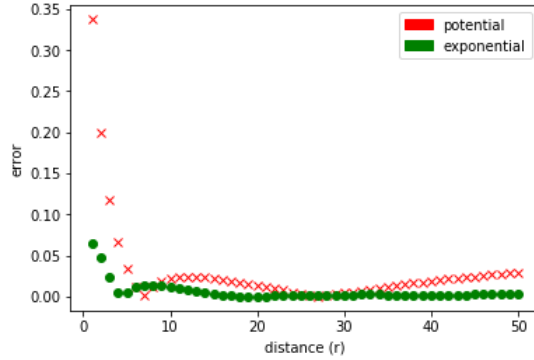


Imagen 2.11.

Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones.  $\beta = 0,85 * \frac{1}{2,26918}$ .

❖ Por encima del punto crítico  $\beta > \frac{1}{2,26918}$

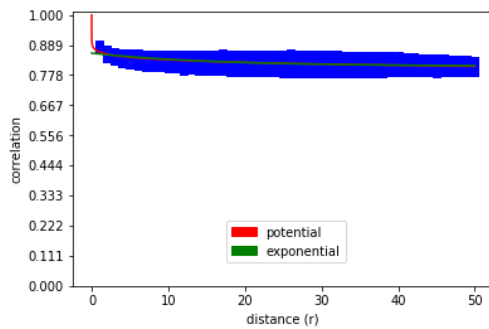


Imagen 2.12.

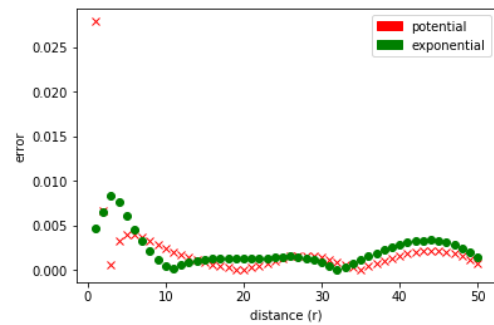


Imagen 2.13.

Este ensayo se ha realizado con una población de 2500 neuronas y 100.000 iteraciones.  $\beta = 1,15 * \frac{1}{2,26918}$ .

La información más relevante que se puede extraer de este apartado es que conforme las simulaciones se realizan más cerca de la temperatura crítica, el modelo potencial aventaja sustancialmente al modelo exponencial. Esto muestra, como se había previsto, que la red trabaja en estado de criticalidad, dentro de la transición de fase. Los resultados de este apartado sirven, además, para poder comparar la red biológica con la red simulada en la parte uno del estudio. Si la red real se ajusta de una forma más fidedigna al modelo potencial se podrá asumir que también trabaja en estado de criticalidad y que las conexiones neuronales junto con sus correlaciones se producen de la misma manera que las simuladas por el modelo.

### Influencia de la magnitud de la red

Para analizar cómo influye el tamaño de la red neuronal sobre la actividad se han realizado nuevas pruebas. En este programa se aplican las relaciones entre temperatura y cuatro magnitudes que son magnetización, susceptibilidad, entropía y capacidad calorífica. Esto se hace ensayando a diferentes temperaturas la misma red neuronal de tamaño  $N$  neuronas, los datos de las diferentes temperaturas se van acumulando para ser representadas todas ellas en los diferentes gráficos frente a las otras magnitudes.

- El magnetismo trata de medir mediante los *spins* cómo se está de cerca del cambio de fase entre la zona ordenada y la zona de comportamiento aleatorio. El concepto se basa en que, por ejemplo, si el comportamiento es eminentemente desordenado la probabilidad de que esté un *spin* en el estado  $+1$  y  $-1$  es aleatorio. En otras palabras, en torno 50% de las veces ocupará una posición u otra. Por otro lado, si el modelo se halla en la región ordenada todos los *spins* tenderán a compartir un mismo estado, en consecuencia, habrá muchas menos paredes de los dominios, *Imagen 1.2*

En la práctica, la magnetización de la red es la probabilidad de que a una temperatura dada la red tenga un determinado estado, comprendido entre 0 y 1.

Así pues, debido a que en la zona desordenada ha habido una cantidad similar de *spins* en una dirección y en otra,  $+1$  ó  $-1$ , el resultado será tendiente a cero de probabilidad (ya que unos anulan a los otros). Por otra parte, si el modelo se halla en una fase ordenada la probabilidad se alejará de cero y tendrá al máximo que es uno, resultado que se verá reflejado en la gráfica.

La manera más sencilla de detectar el punto de criticalidad es fijarse en la zona de criticalidad y localizar en su punto de máxima pendiente la temperatura crítica. En esta ubicación, conforme decae hacia el área desordenada se localiza dicho punto crítico. Tras esto se obtendrá la gráfica que relaciona la temperatura con la susceptibilidad que depende de la magnetización. La susceptibilidad es la derivada de la magnetización. En esta última gráfica se muestra como variando el tamaño de la red, la amplitud de la zona donde la actividad neuronal es mayor varía de una forma significativa, haciendo más rápido el cambio de fase.

- La entropía es una medida muy característica de este tipo de modelos y probablemente una de las más importantes. La gráfica que mide el pico entropía relaciona la temperatura con la incertidumbre de la fuente de información, en otras palabras, indica en que rango de temperaturas la cantidad de información transferida es máxima según la Ecuación 2.4. Esto se realizará con la fórmula de Shannon, Ecuación 2.4., que se usará para el cálculo de la entropía y la capacidad calorífica.

Ecuación 2.4.

$$S = -p(s) \log_{10}(p(s))$$

### Código

La primera parte del código pasa por definir la extensión de la red neuronal a analizar. Es por ello por lo que, para poder analizar las diferentes magnitudes de temperatura, de una red neuronal habrá que implementar cambios en el código. La diferencia más sustancial con el programa principal de ensayo de la parte uno es que este código de simulación neuronal estará incluido dentro de un bucle que analiza el mismo tamaño de red para diferentes temperaturas. Para cada simulación hace la media de todas las interacciones neuronales y así obtener un único valor que es el estado general de la red que variará como se ha referido antes, entre 0 y 1.

La razón de dar un único valor para cada simulación es que a cada temperatura se le asignará ese valor para poder representar en una gráfica todas las temperaturas frente a sus medias de interacción neuronal. Este valor de interacción neuronal o actividad es representativo, en otras palabras, es un valor que no es por sí mismo el estado de ninguna de las neuronas de forma exacta, sino que da una imagen de cómo se está comportando la red. Si el valor se encuentra muy próximo a uno será una red muy rígida con una T baja ya que tendrá muchas de sus conexiones próximas también a este valor. Si por el contrario estuviera muy cercana al valor nulo sería una red casi completamente aleatoria con valores próximos a cero en todas sus neuronas.

El porqué de que no se haya incluido la comparación de varias redes de diferente tamaño dentro del programa con un bucle en el cual se dieran valores y se carguen los archivos de N tamaños manualmente es debido a que el coste computacional sería muy elevado, hablando en términos de varios días de procesamiento.

Tras la obtención de los datos de ensayo, se pasa a su uso para el cálculo de las diferentes gráficas. Para ello se calcula un polinomio que ajuste esta curva de datos de la forma más fidedigna posible. Sin embargo, debido a que son muestras de tamaño finito calculados independientemente hacen que en la gráfica aparezcan picos que son filtrados por el polinomio que refleja una curva mucho más suave.

Es por esto por lo que interesa tener una gran cantidad de datos disponible. Este polinomio representa las probabilidades, como otro objeto de estudio de este apartado es obtener la derivada de la magnetización, es decir, la susceptibilidad se pasará a diferenciar la ecuación y generar un nuevo polinomio. El nuevo polinomio se va a analizar de forma similar al anterior con el mismo número de datos y mismos parámetros. Tras ello se utilizan las probabilidades



almacenadas para representar, a través de la ecuación de Shannon, la gráfica de la entropía y la capacidad calorífica. La entropía es la medida de la energía acumulada. En otras palabras, la entropía aumenta conforme la red se vuelve más aleatoria como si se tratara del cambio de fase sólida a gas de un elemento, de fase ordenada a la contraria, como si se generara un aporte de energía a la red. La capacidad calorífica ilustra donde la variación de entropía es mayor, es decir, el pico de capacidad calorífica se da también en torno al punto crítico.

## Resultados

Los resultados muestran lo esperado en teoría, al aumentar la temperatura disminuye la probabilidad de que el estado general de la red sea de carácter rígido. La red tiende cada vez más a una situación de aleatoria, más entrópica, atravesando un estado intermedio de transición.

En la gráfica de la susceptibilidad vemos el pico lógico debido al drástico cambio de comportamiento en la transición de fase entre la situación de máxima y mínima rigidez. Lo característico de este tipo de comportamiento en la zona de criticalidad es que la temperatura crítica se encuentra en el punto medio, que es la zona de actividad neuronal idónea donde se aplican las teorías de las redes neuronales de Ising. Otra característica importante es que cuanto más crece la red neuronal en aspectos de volumen (número de neuronas), más pendiente aparece en la zona de cambio entre un tipo de comportamiento y otro. Consecuentemente el pico que da nombre a los modelos de máxima entropía en la gráfica de la capacidad calorífica y la susceptibilidad se vuelve más pronunciado y estrecho. La conclusión que se podría inferir es que las redes de un tamaño considerable tienen una mayor capacidad de aprendizaje, como es lógico, pero hay que tener más precisión a la hora de controlar la temperatura crítica para encontrarse en el punto óptimo de funcionamiento.

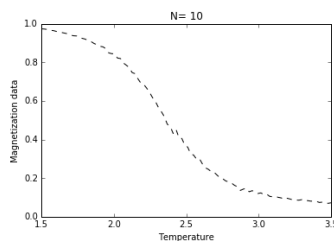


Imagen 2.14.

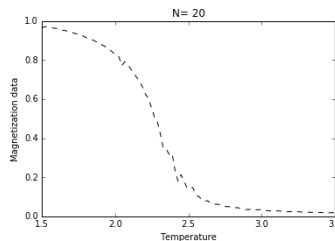


Imagen 2.15.

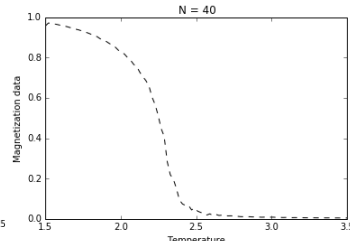


Imagen 2.16.

*Gráfica de probabilidad de estados con una población de 10, 20 y 40 neuronas. Valores de 100.000 iteraciones. Como se muestra en las gráficas, se diferencia de una manera clara la zona de criticalidad donde se produce el cambio de un estado ordenado a otro aleatorio.*

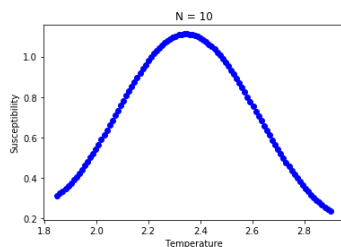


Imagen 2.17.

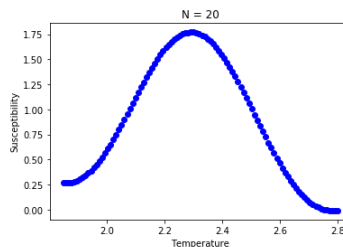


Imagen 2.18.

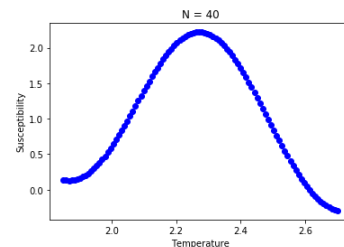


Imagen 2.19.

*A continuación, se muestran las gráficas de la derivada, es decir, la susceptibilidad. El pico se da en torno a la temperatura crítica. Hay que*

*recaltar la influencia del tamaño de la red. Cuanto mayor sea, menor es el rango de temperaturas dentro del pico de susceptibilidad. El rango del eje de abscisas varía de una figura a otra para poder ver el pico de susceptibilidad completo. Para sendos casos de 10, 20 y 40 neuronas.*

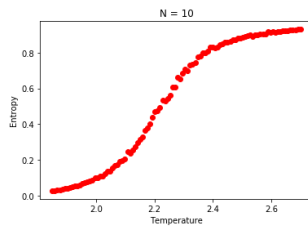


Imagen 2.20.

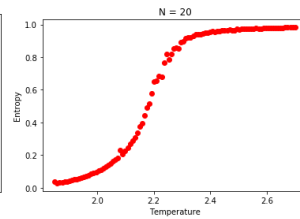


Imagen 2.21.

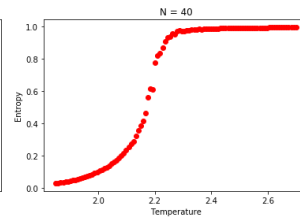


Imagen 2.22.

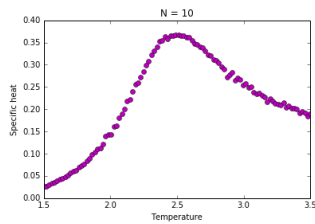


Imagen 2.23.

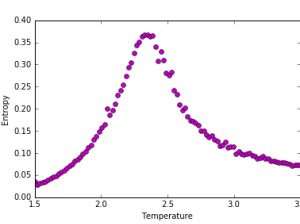


Imagen 2.24.

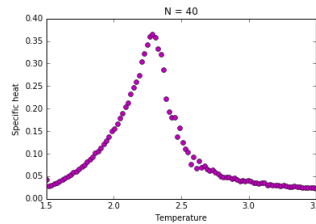


Imagen 2.25.

*Gráfica de la capacidad calorífica. Para sendos casos de 10, 20 y 40 neuronas. Esta gráfica al ser la derivada de la entropía tendrá su máximo donde el cambio de entropía sea máximo y por ello donde el intercambio de información también lo sea, como se ha explicado anteriormente.*

Para concluir cabe destacar entre los resultados obtenidos los picos en la capacidad calorífica y la susceptibilidad. Ambos se sitúan en torno al punto crítico, de ello se puede deducir que es la zona en la que la red es más propensa a adquirir información y procesarla de una manera más eficaz. En otras palabras, cuando la red trabaja a la temperatura crítica aprende de una manera más rápida y responde con una mayor precisión.

## Segunda parte

A continuación, se analizará un caso real de red biológica. Dicha red procede de un conjunto de datos neuronales de la actividad y picos de interacción neuronal de ratones para experimentación. Estos picos neuronales son llamados *spikes* y se organizan en los archivos de la siguiente manera:

La actividad fue analizada y almacenada a lo largo de 34 días *in vitro* (DIV) y 35 cultivos diferentes (CULT), el estado de activación de las neuronas era controlado mediante diferentes electrodos que transmitían la información. Los archivos están organizados mediante las iniciales en mayúsculas que se han descrito anteriormente, así se puede seleccionar cultivos de diferentes días dependiendo de si queremos actividad inicial de los primeros ensayos o de cuando ya se está desarrollando una actividad neuronal plena. En el caso que nos ocupa se ha usado datos de actividad neuronal de mediados de la investigación porque aseguran un estado estacionario de la red, es decir, una actividad normal.

Los datos fueron obtenidos de CRCNS.org: <http://dx.doi.org/10.6080/K0PC308P>, aunque también ha sido referenciado en la bibliografía.

### Código

El código se compone de diferentes partes, cuatro en total. La primera es la adaptación de los datos y la implementación de la información en Python.

Posteriormente hay que filtrar la información ya que aparte de las matrices de actividad neuronal, vienen datos tales como la información de la investigación, tipo de neuronas, etc. Para ello hay que detectar el número de picos de interacción neuronal, extraer el número de datos y a partir de ahí crear una matriz con todos los datos existentes.

Tras la obtención de los datos “en limpio”, se trata de analizar la información de las interacciones neuronales como se ha venido haciendo en la parte interior, sustituyendo la interacción que nos da el programa por unos datos que son picos de actividad reales llamados *spikes*. Para concluir, se representará en diferentes diagramas la forma y características de la actividad neuronal, de esta manera podrá ser comparada con los datos de la simulación del programa de la primera parte.

### Resultados

#### *Picos de actividad e interacción*

En lo referente a la actividad y sus picos, *spikes*, hay que subrayar que conforme avanzan los días de estudio en probetas, las zonas y tiempos donde se registra actividad van quedando más diferencialmente marcadas. Se puede apreciar tanto en la matriz como en la gráfica de los picos de actividad cómo van tendiendo a iluminarse zonas segmentada de mayor área. Por el contrario, la dispersión de los *spikes* va disminuyendo. Esto indica que va tendiendo hacia el estado ordenado partiendo de uno totalmente entrópico, sin alcanzar un estado totalmente rígido. También se ve reflejado en la escala de los valores de la matriz de actividad, al aumentar

los días *in vitro*, crece el rango de valores. En otras palabras, el valor máximo de la interacción neuronal aumenta, lo cual comporta una mayor transmisión de información.

Por ello, como se ha mencionado anteriormente, para un estudio de la actividad neuronal de un ser vivo es más concluyente el análisis de un cultivo que lleve ya unos días de actividad e incubación. Aun con todo, se ve claramente que los valores máximos de dicha actividad están lejos de alcanzar el valor unidad, valor máximo. La posible razón es que son cultivos que se han extraído y almacenado en unas condiciones concretas que no son las naturales de dicha red neuronal. A esto hay que sumarle que se activan durante un periodo de tiempo que no puede superar entre los 34-36 días *in vitro* y que los rangos de exposición de la red a estímulos son de un periodo de tiempo no superior a entorno 180000 unidades que es equivalente a 9000ms.

De estos rangos de exposición o periodos de actividad también se puede inferir que conforme pasa el tiempo la actividad neuronal incrementa, aunque no de una forma brusca.

Otro punto a tener en cuenta es que la zona de más interacción, que contiene más picos de actividad es la zona más próxima a donde se realizan los estímulos. Con mucha diferencia estas neuronas realizan un intercambio de información elevado que demuestra como en la parte primera de la investigación que sigue una tendencia potencial. Esto se ilustra a continuación.

❖ Cultivo 3

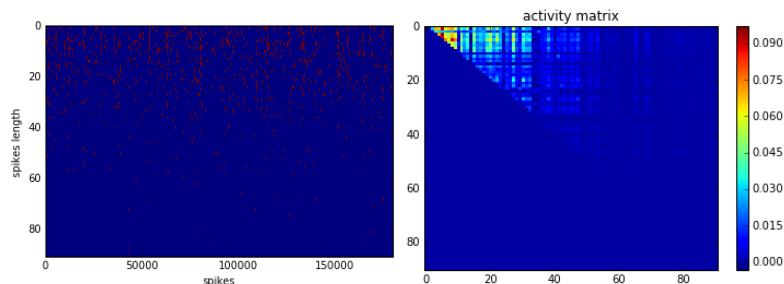


Imagen 3.1.

Imagen 3.2.

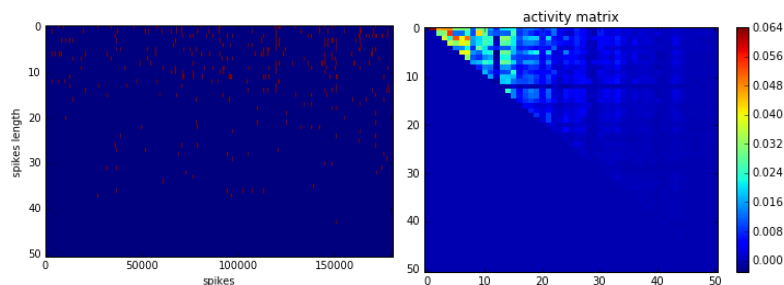


Imagen 3.2.

Imagen 3.4.

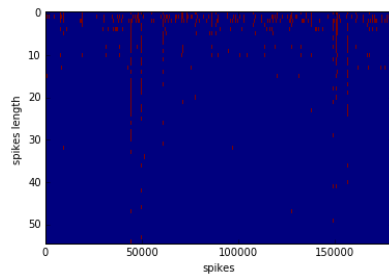


Imagen 3.5.

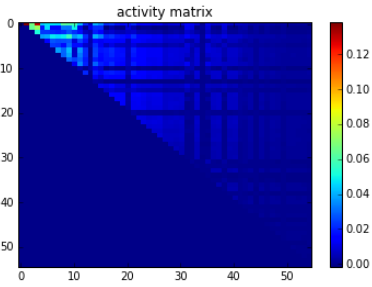


Imagen 3.6.

*Imágenes de los picos de actividad a largo del tiempo de ensayo y la matriz de interacción neuronal. Los ensayos están hechos en los días de incubación (DIV) 5, 24,35. Dependiendo de los datos usados la matriz de actividad puede variar ligeramente. Sin embargo, se ha comprobado que los resultados obtenidos son más claros conforme se escogen muestras que llevan una cantidad mayor de días in vitro.*

#### ❖ Cultivo 5

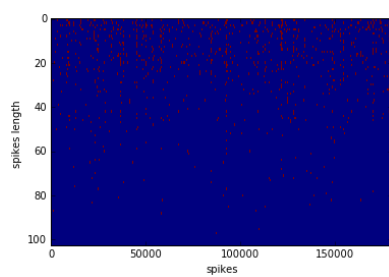


Imagen 3.7.

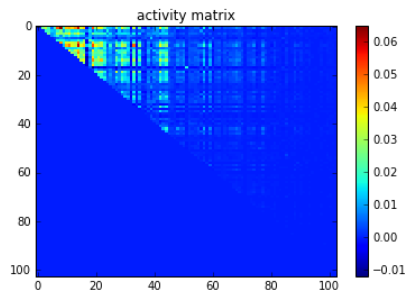


Imagen 3.8.

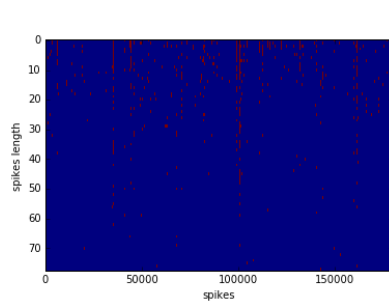


Imagen 3.9.

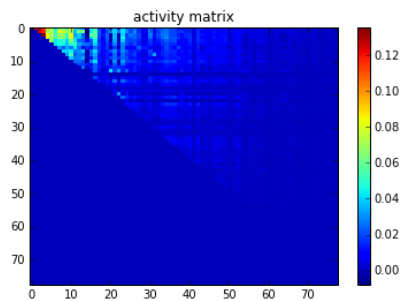


Imagen 3.10.

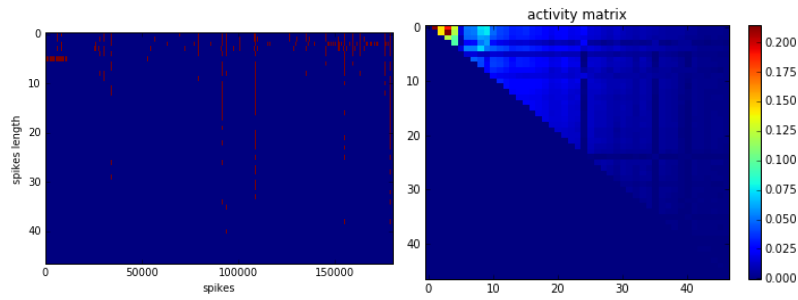


Imagen 3.11.

Imagen 3.12.

*Imágenes de los picos de actividad a largo del tiempo del ensayo y la matriz de interacción neuronal. Los ensayos están hechos en los días de incubación (DIV) 5, 27,35. Los resultados son muy similares a los del cultivo anterior a lo que valores de actividad se refiere. La localización de los picos de actividad o spikes varía suceso normal ya que al ser otro cultivo responderá en otros instantes de tiempo a los impulsos.*

### *Ajuste del modelo*

Como se realizó en la primera parte se verá cual es el modelo que mejor se aproxima a esta red. En el primer caso se pudo comprobar como una red en torno al punto de criticalidad, dentro del rango de temperatura crítica en la transición de fase, la curva potencial se aproxima de manera más exacta que la exponencial. Lo que se quiere demostrar en este estudio es que una red neuronal real trabaja dentro de este estado de transición y por tanto en el punto crítico.

A continuación, se muestran los ajustes y sus sendos errores, dependiendo los resultados y de cómo se ajustan a los datos empíricos se aceptaría una ecuación u otra. Determinando así en el rango de temperaturas dentro del cual se ubica la red.

Como se ha comentado en la actividad neuronal, la red trabaja con interacciones muy bajas, acumulando casi toda su actividad (en torno al 95% de la interacción neuronal) por debajo del 0.4. Consecuentemente ambas ecuaciones, potencial y exponencial, reproducen de una manera fidedigna la actividad neuronal debido a la falta de datos o una curva más robusta. Para solventar este contratiempo cobra relevancia el cálculo del error que a simple vista es difícil de interpretar, pero a la hora de mostrar resultados arroja una conclusión clara. El modelo potencial es sustancialmente mejor con una exactitud entre 5 y 10 veces mayor que el modelo exponencial.

❖ Cultivo 3

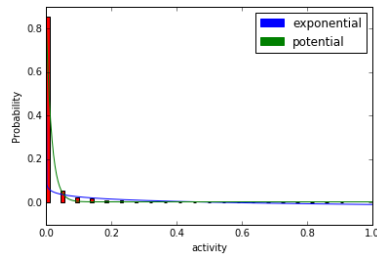


Imagen 3.13.

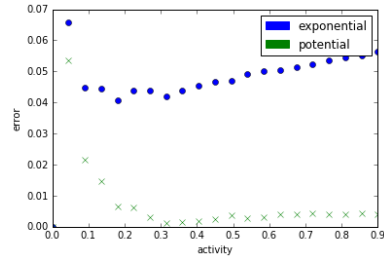


Imagen 3.14.

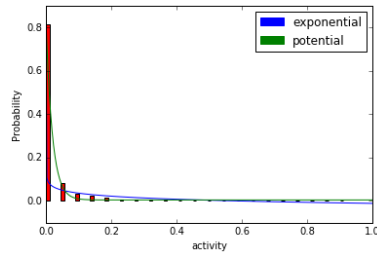


Imagen 3.15.

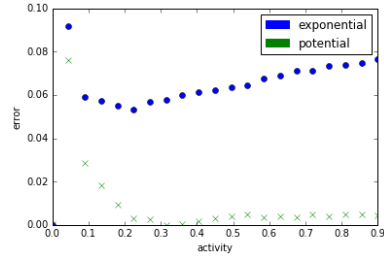


Imagen 3.16.

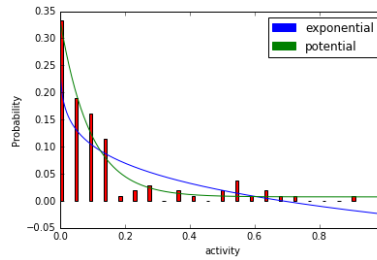


Imagen 3.17.

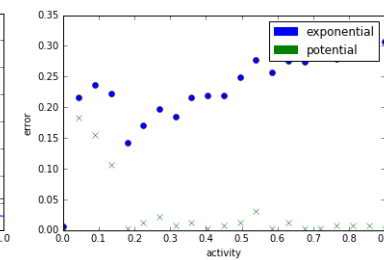


Imagen 3.18.

*Gráficas de la probabilidad del Estado neuronal, rango de actividad. Sobre ella la curva de ajuste potencial y exponencial. A la derecha el error cometido por sendos ajustes. Los ensayos están hechos en los días de incubación (DIV) 6, 24,35. Como se esperaba la diferencia del error cometido por la gráfica potencial y exponencial aumenta conforme el cultivo de estudio lleva más días in vitro respondiendo a estímulos,*

❖ Cultivo 5

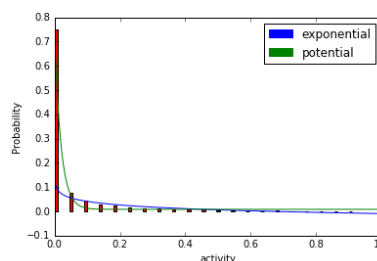


Imagen 3.19.

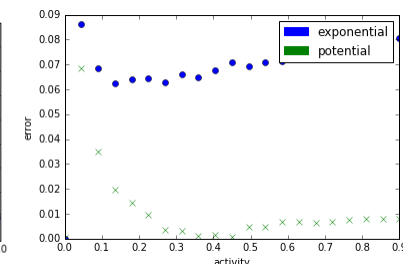


Imagen 3.20.

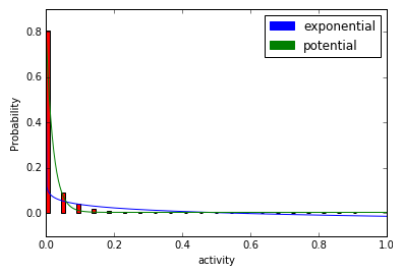


Imagen 3.21.

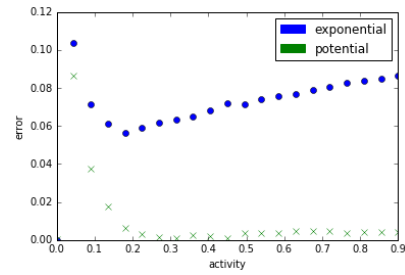


Imagen 3.22.

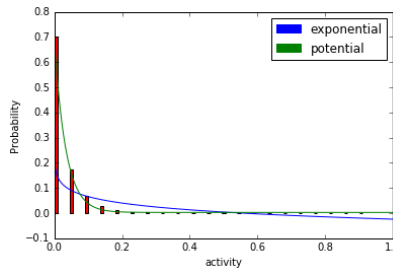


Imagen 3.23.

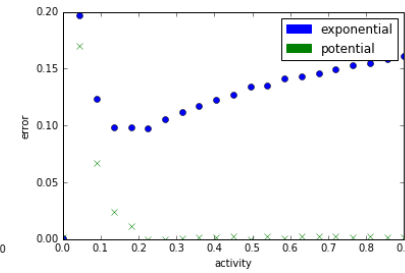


Imagen 3.24.

*Gráficas de la probabilidad del Estado neuronal, rango de actividad. Sobre ella la curva de ajuste potencial y exponencial. A la derecha el error cometido por sendos ajustes. Los ensayos están hechos en los días de incubación (DIV) 6, 27,35. La situación es similar al cultivo anterior, la potencial mejora su ajuste respecto a los daros conforme aumentan los días de incubación.*

Como conclusión para este apartado, se ha analizado como se ajustan los modelos potencial y exponencial a los datos biológicos. Los resultados han sido satisfactorios, el ajuste potencial se ha aproximado de una manera más precisa a los datos biológicos, pudiendo aceptar como válido que la red neuronal del mamífero estudiado trabaja en un estado de criticalidad.



## Coste computacional y otras dificultades

- El **coste computacional** ha supuesto una traba a la hora de realizar los ensayos. Sobre todo, en la parte correspondiente al entrenamiento neuronal los requerimientos de procesamiento eran altos por lo que se necesita de ordenadores potentes, de última generación para hacer los cálculos en un tiempo relativamente corto. Esto limita el rango de redes neuronales que podemos examinar no pudiendo analizar resultados en redes mayores de 2500 neuronas (50x50).

Otro problema es el número de iteraciones que se pueden realizar en un tiempo razonable. Cuantas más iteraciones podamos realizar más exactos serán los modelos en la parte de entrenamiento. Esto es debido a que las redes de mayor tamaño (50 × 50) tardan bastante más en converger en lo que se refiere a los resultados de actividad y correlaciones. Ello implica que, para tener resultados claros sin mucha dispersión, es necesario realizar al menos 100.000 iteraciones y sería bastante deseable alcanzar el millón, aunque con la tecnología disponible supondría un coste computacional excesivo. A continuación, se adjunta tabla de tiempos medios del procesamiento de los diferentes programas realizados.

Código	Tiempo promedio de coste computacional, $T_M$
Entrenamiento/N= 50x50/ 10000 iteraciones	6h 30 min
Análisis de entropía/N=40x40/100000 iteraciones	5h
Análisis red neuronal biológica	2 min
Análisis del error	30 s

Tabla 4.1.

*Cómo se ha mencionado, lo más notable de la tabla es la sustancial diferencia entre el tiempo de cálculo entre los programas que llevan incluido una parte de entrenamiento neuronal para conformar la red. El programa que capta directamente los datos de redes neuronales reales al estar exento de estos cálculos es mucho más rápido.*

- Los **archivos de las redes neuronales** vivas no contenían exactamente el mismo número de muestras. Iban variando su contenido y también la calidad o capacidad de análisis de las muestras, había mucha variabilidad de resultados dependiendo del estado de la red y el cultivo. Por estado de la red se quiere dar a entender que había cultivos en los que determinadas muestras no seguían el patrón que mostraban las mismas muestras en diferentes cultivos, es decir, muestras en el mismo día de incubación, pero perteneciente a otras partes de la red neuronal del ratón. Además, había que escoger bien los días *in vitro* porque también variaba mucho la actividad neuronal, aunque la tendencia general era que la actividad aumentaba conforme aumentaban los DIV.

## Conclusión

Tras los estudios realizados, se ha conseguido arrojar luz sobre el análisis de la información dentro de una red neuronal. Cómo dicho procesamiento, que es la interacción entre pares de neuronas, se ve afectado por diferentes variables como la distancia y el punto de trabajo de la red.

A continuación, recopilamos los resultados obtenidos en cada apartado del estudio de cada apartado y los resultados obtenidos en cada uno de ellos:

- En la primera parte se ha simulado una red neuronal a diferentes temperaturas. Se ha podido concluir que, en la zona de criticalidad, donde se localiza el punto crítico, las correlaciones varían con la distancia; incluso neuronas lejanas están correlacionadas sin que suponga que toda la red está completamente sincronizada. Además de que, como se ha mencionado, la correlación entre neuronas nunca decae a cero, por lo que dos neuronas de la red por muy separadas que estén intercambian información.

También se ha estudiado la aproximación potencial y exponencial de la red, en otras palabras, se ha ajustado la distribución de las correlaciones por estas dos curvas. El ajuste potencial es el que mejor modela una red neuronal que trabaje en estado de criticalidad. Se ha concluido que en la fase de transición la red trabaja en dicho estado, mostrando que una distribución potencial de correlaciones es un indicativo sencillo de medir que un sistema se encuentra cerca del punto crítico. A partir de la utilización del modelo potencial en la simulación, se hará lo mismo para la red biológica, si esta red también se ajusta de una manera eficaz por el modelo potencial se podrá asumir que ambas redes se comportan de una manera muy parecida a la hora de intercambiar información.

Por último, se han estudiado diferentes propiedades de la red neuronal, como son la magnetización, la susceptibilidad, la entropía y la capacidad calorífica. Centrándonos en las más importantes, la capacidad calorífica mide cómo de sensible es la red ante estímulos externos, siendo un indicador de la capacidad de procesamiento de información del sistema de la red. Como se muestra en las gráficas, el pico de la capacidad calorífica, que es el que indica dónde se maximiza dicho intercambio de información, se ubica en torno al punto crítico.

- En el segundo apartado se han analizado los datos de una red neuronal biológica. La razón de dicho estudio es comprobar si una red real trabaja en estado de criticalidad. Ello implicaría que los resultados y propiedades obtenidas de las simulaciones de redes artificiales serían trasladables a la actividad neuronal de seres vivos para investigaciones futuras. Finalmente se ha constatado que el modelo potencial es el que ajusta de una manera más fidedigna los datos de estudio, esto sustenta la afirmación de que la red neuronal del mamífero estudiado se encuentra en el estado de criticalidad. Se puede afirmar que el modelo podría usarse para reproducir comportamientos neuronales reales.

Las diferentes conclusiones extraídas en la investigación se pueden enumerar en los siguientes puntos:

- **El punto crítico**, que es regido por la temperatura crítica, es la región que da un comportamiento de la red más eficaz. Esto se debe a que, aunque la interacción decae con la distancia, en esta zona nunca es nula y se comporta como un cerebro o red neuronal viva, en la que cualquier neurona tiene capacidad de comunicarse con cualquier otra de la misma red. Fuera del punto crítico el comportamiento tiende a concurrir hacia dos tipos de estados. En uno de ellos la red responde de una manera muy rígida, obteniendo siempre el misma '*output*' ante cualquier estímulo, '*input*'. Por otro lado, puede asimilarse a un sistema estocástico fuertemente aleatorio en el cual la respuesta es muy azarosa, sin apenas relación con el estímulo.
- La **distancia**, sobre la que se ha estado hablando a lo largo del texto es capital para entender una red de dos o más dimensiones. Es importante ya que en situaciones reales, como las que tratan de simular los modelos aquí realizados, se aprecia que las neuronas intercambian más información con otras neuronas que sean colindantes. En el estudio de este intercambio, que aquí se menciona como actividad y correlación neuronal, se ha comprobado que sigue dos tipos de reglas de dependencia con la distancia. Estas son la potencial y la exponencial. La potencial es la que modela la actividad en la zona de criticalidad que es donde conviven las dos fases de comportamiento, rígido y aleatorio. Esta zona contiene el punto crítico y consecuentemente la zona de actividad más parecida a la real. El uso de estas dos funciones sobre datos reales da la clave para saber si se pueden trasladar los modelos neuronales creados por ordenadores a las respuestas cognitivas de seres vivos.
- El análisis de las redes neuronales biológicas es la parte con la que concluye el estudio. Tras haber obtenido las ecuaciones que modelan el comportamiento de las redes neuronales entrenadas en el algoritmo dentro y fuera de la temperatura crítica se ha estudiado como se asemejan unas redes y otras. La conclusión ha sido satisfactoria ya que se ha constatado que el modelo potencial es mucho más preciso a la hora de predecir la interacción neuronal en los picos de actividad del cerebro del mamífero objeto de estudio. Por ello se puede intuir que la red neuronal del ser vivo trabaja en un estado de criticalidad o próximo a éste.

Como resumen hay que decir que el algoritmo Monte Carlo tiene sus limitaciones. El elevado tiempo de procesamiento es algo común en este tipo de programas y modelos que se irá solventando con el tiempo. Pese a esto, el modelo de probabilidad se ajusta de una manera precisa a las muestras neuronales, lo cual se ve reflejado en los resultados, sirviendo por lo tanto para el modelado de redes neuronales a escala real. Tras este estudio, queda por delante realizar nuevas investigaciones para construir un algoritmo propio a partir de los datos obtenidos en este proyecto a través de un modelo de Ising cinético con un método de descenso de gradiente.

En un contexto general este estudio puede aplicarse a otros campos.

En medicina, más específicamente en neurociencia, ayudaría a sustentar la información que se tiene actualmente propuesta entre otros por el neurólogo Rodolfo Llinás, la cual rebate que las diferentes partes del cerebro trabajan de una forma aislada. La correlación que existe en la red entre todos los pares de neuronas, aunque haya una distancia considerable entre ellas, podría hacer cambiar el modo en el que se estudia la actividad cerebral.

Dentro del rango de la temperatura crítica se ha comprobado que las redes son más sensibles a los estímulos, porque la susceptibilidad y actividad neuronal son mayores. Esto podría tener implicaciones en como el sistema neuronal capta nuestro entorno y como se relaciona el punto de criticalidad con fenómenos más complejos como la consciencia. Teoría propuesta por (*Ouellette Laboratory, 2014*).

## Bibliografía

- Nicholas M. Timme, Najja Marshall, Nicholas Bennett, Monica Ripp, Edward Lautzenhiser and John M. Beggs. (2016). Spontaneous spiking activity of thousands of neurons in rat hippocampal dissociated cultures. CRCNS.org: <http://dx.doi.org/10.6080/K0PC308P>
- David H. Ackley Geoffrey E. Hinton J. Sejnowski. (1985). A learning algorithm for boltzmann machines, Computer science department and biophysics department the Johns Hopkins University
- Gasper Tkacik, Elad Schneidman, Michael J. Berry , William Bialeka. Joseph Henry. (December 30, 2009 (accessed: Nov 16 2017)). Spin glass models for a network of real neurons, Laboratories of Physics, Department of Molecular Biology, Lewis–Sigler Institute for Integrative Genomics, and Princeton Center for Theoretical Science, Princeton University. [https://www.researchgate.net/publication/45892740\\_Spin\\_glass\\_models\\_for\\_a\\_network\\_of\\_real\\_neurons](https://www.researchgate.net/publication/45892740_Spin_glass_models_for_a_network_of_real_neurons).
- Aguilera, M., Alquezar, C., and Izquierdo, E.J. (2017). Signatures of criticality in a maximum entropy model of the C. elegans brain during free behaviour. Proceedings of the 14th European Conference of Artificial Life. Lyon, France.
- Beggs, J. M. (2008). The criticality hypothesis: How local cortical networks might optimize in formation processing. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 366(1864), 329-343.
- Bialek, W., Cavagna, A., Giardina, I., Mora, T., Silvestri, E., Viale, M., & Walczak, A. M. (2012). Statistical mechanics for natural flocks of birds. Proceedings of the National Academy of Sciences, 109(13), 4786-4791.
- Botcharova, M., Farmer, S. F., & Berthouze, L. (2014). Markers of criticality in phase synchronization. Frontiers in Systems Neuroscience, 8.
- Quiroga, R.Q., Z. Nadasdy, and Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Computation, 2004. 16: p. 1661-1687.
- Durbin, R. M. (1987). Studies on the development and organisation of the nervous system of caenorhabditis elegans.
- Wornatlas, Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology, 117(4), 500.
- Mora, T., Walczak, A. M., Bialek, W., & Callan, C. G. (2010). Maximum entropy models for antibody diversity. Proceedings of the National Academy of Sciences, 107(12), 5405- 5410.
- Nguyen, J. P., Shipley, F. B., Linder, A. N., Plummer, G. S., Liu, M., Setru, S. U., . . . Leifer, A. M. (2016). Whole-brain calcium imaging with cellular resolution in freely behaving caenorhabditis elegans. Proceedings of the National Academy of Sciences, 113(8), E1081.

- Ouellette, J. (2014). A fundamental theory to model the mind. *Quanta Magazine*. Stern, Y. (2002).
- Yaakov Stern. (2003). What is cognitive reserve? theory and research application of the reserve concept. *Journal of the International Neuropsychological Society*, 8(03), 448-460. G.H. Sergievsky Center, The Taub Institute, Department of Neurology, Columbia University College of Physicians and Surgeons, New York, NY 10032, USA.
- Tkačik, G., Mora, T., Marre, O., Amodei, D., Palmer, S. E., Berry, M. J., & Bialek, W. (2015 Sep 15). Thermodynamics and signatures of criticality in a network of neurons.
- Tyrcha, J., Roudi, Y., Marsili, M., & Hertz, J. (2013). The effect of nonstationarity on models inferred from neural data. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(03), P03005 Mora, T., & Bialek, W. (2011).
- Thierry Mora · William Biale. (12 December 2010). Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2), 268-302.
- N. M. Timme, N. Marshall, N. Bennett, M. Ripp, E. Lautzenhiser, and J. M. Beggs . (2016). Criticality maximizes complexity in neural tissue, *Frontiers in Physiology*, 7 (425). <http://dx.doi.org/10.3389/fphys.2016.00425>.

## Anexos

### Código de simulación de la red neuronal

```

from __future__ import division
import numpy as np
from numpy.random import rand
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from scipy.optimize import curve_fit
import math
import seaborn as sns
N = 50
NN=np.zeros((N,N))
## monte carlo moves
def mcmove(config,beta):
    for i in range(N):
        for j in range(N):
            a = np.random.randint(0, N)
            b = np.random.randint(0, N)
            s = config[a, b]
            nb = config[(a+1)%N,b] + config[a,(b+1)%N] + config[(a-1)%N,b] + config[a,(b-1)%N]
            cost = 2*s*nb
            if cost < 0:
                s *= -1
            elif rand() < np.exp(-cost*beta):
                s *= -1
            config[a, b] = s

    return config

def initialstate(N): # generates a random spin config
    state = 2*np.random.randint(2, size=(N,N))-1
    return state

#ising parameters
config = initialstate(N)
mcSteps = 100000
NNN=np.zeros((N*N,N*N))

temperature_grade=[1]
critic_temperature=2/np.log(1+np.sqrt(2))
print('critic temperature=',critic_temperature)
for i in range(len(temperature_grade)):
    print('déjame trabajar')
    beta=temperature_grade[i]*1/critic_temperature
    for i in range(mcSteps):
        mcmove(config, beta)
    a=0

```

```
b=0
c=0
d=0

z=0
y=0
x=0

A = np.asarray(config).reshape(-1)

B=np.transpose(A)
NNN=NNN+np.outer(B,A)

NNN=NNN/mcSteps

#distance
DISTHOR=np.zeros((N*N,N*N))
DISTVER=np.zeros((N*N,N*N))
w=0
v=0
x=0
row=0
for i in range(len(NNN)):
    for j in range(len(NNN)):
        DISTHOR[i,j]= abs(w - row)
        v=v+1
        if (v==N):
            w=w+1
            v=0
        w=0
        x=x+1
        if (x==N):
            row=row+1
            x=0

w=0
v=0
x=0
column=0
for i in range(len(NNN)):
    for j in range(len(NNN)):
        DISTVER[i,j]= abs(w - column)
        v=v+1
        w=w+1
        if (v==N):
            w=0
            v=0
```



```
w=0
x=x+1
column=column + 1
if (x==N):
    column=0
    x=0

#adjusting the net to periodic matrix
for i in range(len(DISTHOR)):
    for j in range(len(DISTHOR)):
        if ( -1* DISTHOR[i,j]%N<DISTHOR[i,j]):
            DISTHOR[i,j]=-1*DISTHOR[i,j]%N
for i in range(len(DISTVER)):
    for j in range(len(DISTVER)):
        if ( -1* DISTVER[i,j]%N<DISTVER[i,j]):
            DISTVER[i,j]=-1*DISTVER[i,j]%N

DIST=DISTHOR+DISTVER

#triang sup matrix (both distance and correlation matrix)
for i in range(len(NNN)):
    for j in range(len(NNN)):
        if (i>j) :
            NNN[i,j]=0
            DIST[i,j]=0

plt.show()
#print(NNN)
#print(DIST)
plt.imshow(NNN)

plt.title('corelation matrix')
plt.colorbar()
plt.show()
plt.imshow(DIST)
plt.title('distance matrix (r)')
plt.colorbar()
plt.show()

#plots, matrix and equation fit

C = np.asarray(NNN).reshape(-1)
D = np.asarray(DIST).reshape(-1)

#plt.plot(D, C,'x')

plt.show()
```

```

j=0
for i in range(len(C)):
    if((C[i]!=0)) :
        j=j+1
poligraph=np.zeros(j)
distpoligraph=np.zeros(j)
j=0
for i in range(len(C)):
    if((C[i]!=0)) :
        poligraph[j]=C[i]
        distpoligraph[j]=D[i]
        j=j+1

j=0
for i in range(len(poligraph)):
    if((poligraph[i]<0.98)) :
        j=j+1
poli=np.zeros(j)
distpoli=np.zeros(j)

j=0
for i in range(len(poligraph)):
    if((poligraph[i]<0.98)) :
        poli[j]=poligraph[i]
        distpoli[j]= distpoligraph[i]
        j=j+1
plt.plot(distpoli, poli, 'x')
x = poli
y = distpoli

plt.show()
xdata=distpoli
ydata=poli
plt.plot(distpoli, poli,'cs')
plt.xlabel('distance (r)')
plt.ylabel('correlation')
f=np.linspace(0,1,10)
plt.yticks(f)
plt.show()
xdata=distpoli
ydata=poli

#error
#

```

---

```

k=int(max(xdata) )

```

```

l=len(ydata)

ajusted_correlation=np.zeros(k)
ajusted_distance=np.linspace(1,k,k)

errorfit=np.zeros(k)
error=np.zeros(k)
criticerrorfit=np.zeros(k)
criticerror=np.zeros(k)

w=0
cont=0
for i in range(k):
    for j in range (len(ydata)):
        if (xdata[j]== i+1):
            w=w+ydata[j]
            cont=cont+1
    ajusted_correlation[i]=w/cont
    cont=0
    w=0

def graph(formula, x_range, t):
    x = np.array(x_range)
    y = eval(formula)
    plt.plot(x, y, color=t)
#     plt.show()

T=0
Y=0
Z=0
plt.plot(distpoli, poli, 'bs')

def funcritic(x, a, b, c):
    global T,Y,Z
    T=a
    Y=b
    Z=c
    return a*1/(x**(-b)) +c
popt, pcov = curve_fit(funcritic, distpoligraph, poligraph)
graph(' T*1/(x**(-Y))+Z ', np.linspace(0, N,num=N**3), 'r')
red_patch = mpatches.Patch(color='red', label='critic point')

```

```

for i in range(k):
    criticerror[i]=np.absolute((T*1/((i+1)**(-Y))+Z)-ajusted_correlation[i])

A=0
B=0
C=0
def func(x, a, b, c):
    global A,B,C
    A=a
    B=b
    C=c
    return a*np.exp(-b * x) +c

popt, pcov = curve_fit(func, distpoligraph, poligraph)

graph(' A*np.exp(-B* x)+C ', np.linspace(0, N,num=N**3), 'g')
green_patch = mpatches.Patch(color='green', label='out of the critic point')
plt.legend(handles=[red_patch,green_patch])
plt.xlabel('distance (r)')
plt.ylabel('correlation')
plt.yticks(f)
plt.show()

for i in range(k):
    error[i]=np.absolute(( A*np.exp(-B*(i+1))+C )-ajusted_correlation[i])

e=0
c=0
cont=0

print(l)

plt.plot(ajusted_distance, criticerror,'x', color='r',label="critic point")
plt.plot(ajusted_distance, error, 'o', color='g',label="out of the critic point")
plt.legend(handles=[red_patch,green_patch])
plt.xlabel('distance (r)')
plt.ylabel('error')
plt.show()

```

## Código de análisis de entropía y magnetización

```

from __future__ import division
import numpy as np
from numpy.random import rand
import matplotlib.pyplot as plt
import scipy.misc as scp
from scipy.optimize import curve_fit
import sympy as s
import math
N = 40
NN=np.zeros((N,N))
# monte carlo moves
def mcmove(config,beta):
    for i in range(N):
        for j in range(N):
            a = np.random.randint(0, N)
            b = np.random.randint(0, N)
            s = config[a, b]
            nb = config[(a+1)%N,b] + config[a,(b+1)%N] + config[(a-1)%N,b] + config[a,(b-1)%N]
            cost = 2*s*nb
            if cost < 0:
                s *= -1
            elif rand() < np.exp(-cost*beta):
                s *= -1
            config[a, b] = s

    return config

def calcEntropy(config):
    '''Energy of a given configuration'''
    energy = 0
    for i in range(len(config)):
        for j in range(len(config)):
            S = config[i,j]
            nb = config[(i+1)%N, j] + config[i,(j+1)%N] + config[(i-1)%N, j] + config[i,(j-1)%N]
            energy += -nb*S
    return energy/4.

def initialstate(N): # generates a random spin config
    state = 2*np.random.randint(2, size=(N,N))-1
    return state

#ising parameters
config = initialstate(N)
mcSteps = 10000
NNN=np.zeros((N*N,N*N))
MMM=np.zeros((N*N,N*N))
q=100
data=np.zeros(q)

```

```

temperatures=np.linspace(1.5,3.5 ,q)
file_pos=0
for v in range(len(data)):

    temperature=temperatures[v]
    beta=1/temperature
    #main loop
    print('dejame trabajar')
    for i in range(mcSteps):
        mcmove(config, beta)
        energy=calcEntropy(config)
        a=0
        b=0
        c=0
        d=0

    z=0
    y=0
    x=0

    A = np.asarray(config).reshape(-1)
    C= np.asarray(energy).reshape(-1)
    B= np.transpose(A)
    D= np.transpose(C)
    NNN=NNN+np.outer(B,A)
    MMM+=np.outer(D,C)

NNN=NNN/mcSteps
MMM=MMM/mcSteps

#distance
DISTHOR=np.zeros((N*N,N*N))
DISTVER=np.zeros((N*N,N*N))
w=0
v=0
x=0
row=0
for i in range(len(NNN)):
    for j in range(len(NNN)):
        DISTHOR[i,j]= abs(w - row)
        v=v+1
        if (v==N):
            w=w+1
            v=0
    w=0
    x=x+1
    if (x==N):
        row=row+1
        x=0

```

```
w=0
v=0
x=0
column=0
for i in range(len(NNN)):
    for j in range(len(NNN)):
        DISTVER[i,j]= abs(w - column)
        v=v+1
        w=w+1
        if (v==N):
            w=0
            v=0
    w=0
    x=x+1
    column=column + 1
    if (x==N):
        column=0
        x=0
```

```
#adjusting the net to periodic matrix
for i in range(len(DISTHOR)):
    for j in range(len(DISTHOR)):
        if ( -1* DISTHOR[i,j]%N<DISTHOR[i,j]):
            DISTHOR[i,j]=-1*DISTHOR[i,j]%N
for i in range(len(DISTVER)):
    for j in range(len(DISTVER)):
        if ( -1* DISTVER[i,j]%N<DISTVER[i,j]):
            DISTVER[i,j]=-1*DISTVER[i,j]%N
```

```
DIST=DISTHOR+DISTVER
```

```
#triang sup matrix (both distance and activity matrix)
for i in range(len(NNN)):
    for j in range(len(NNN)):
        if (i>j) :
            NNN[i,j]=0
            DIST[i,j]=0
            MMM[i,j]=0
```

```
#matrix and equation fit
```

```
C = np.asarray(NNN).reshape(-1)
D = np.asarray(DIST).reshape(-1)
```

```
E = np.asarray(MMM).reshape(-1)
j=0
for i in range(len(C)):
    if( C[i]!=0) :
        j=j+1
poli=np.zeros(j)
distpoli=np.zeros(j)
polienergy=np.zeros(j)
j=0
for i in range(len(C)):
    if( C[i]!=0) :
        poli[j]=C[i]
        distpoli[j]=D[i]
        polienergy[j]=E[i]
        j=j+1

xdata=distpoli
ydata=poli
edata=polienergy

n=len(ydata)
p=0
q=0
for i in range(len(ydata)):
    p+=ydata[i]
    q+=edata[i]

p=p/n
q=q/n
data[file_pos]=p
file_pos+=1

np.save('magnetizationdata20',data)
np.save('entropyondata20',data)

#Magnetization & Susceptibility
#_____

def graph(formula, x_range):
    x = np.array(x_range)
    y = eval(formula)
    plt.plot(x, y)

Probability_list=np.load('magnetizationdata20.npy')
print(Probability_list)
probability_pos=np.linspace(1,2,q)
```



```

plt.plot(temperatures,Probability_list,'k--')
plt
plt.title('N= 0')
opacity=0.5
plt.xlabel('Temperature')
plt.ylabel('Magnetization data')
plt.show()
p=np.polyfit(temperatures,Probability_list,10)
print(p)
A=p[0]
B=p[1]
C=p[2]
D=p[3]
E=p[4]
F=p[5]
G=p[6]
H=p[7]
I=p[8]
J=p[9]

graph('A*x**10+B*x**9+C*x**8+D*x**7+E*x**6+F*x**5+G*x**4+H*x**3+I*x**2+J*x**1',te
mperatures)
opacity=1

plt.xlabel('Temperature')
plt.ylabel('Magnetization')
plt.title('N = 20')
plt.show()

def f(x):
    return A*x**10+B*x**9+C*x**8+D*x**7+E*x**6+F*x**5+G*x**4+H*x**3+I*x**2+J*x**1

couplings=np.zeros(q)

x = s.var ('x')
plt.title('N = 20')

y=A*x**10+B*x**9+C*x**8+D*x**7+E*x**6+F*x**5+G*x**4+H*x**3+I*x**2+J*x**1

couplings=np.zeros(q)

f_derivative=y.diff(x)
f = s.lambdify(x, f_derivative, 'numpy')

couplings=-1*(f(temperatures))
plt.plot(temperatures,couplings)
plt.xlabel('Temperature')
plt.ylabel('Susceptibility')
plt.show()

```

```
#

---

#Entropy & heat capacity

Entropy_list=np.load('entropydata20.npy')

plt.plot(temperatures,np.absolute(Entropy_list-1),'ro')
plt.title('N = 20')
plt.xlabel('Temperature')
plt.ylabel('Entropy')

plt.show()

n=len(Probability_list)
Entropy_list=np.zeros(n)
for i in range(n):
    Entropy_list[i]=-1*(Probability_list[i]*math.log(Probability_list[i]))
plt.plot(temperatures,Entropy_list,'mo')
plt.xlabel('Temperature')
plt.ylabel('Specific heat')
plt.title('N = 20')
plt.show()
```

## Código de análisis de datos biológicos

```
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy import io
from scipy.optimize import curve_fit
from PIL import Image
import matplotlib as mt
import matplotlib.patches as mpatches
#from infoflow import Entropy,MI
#from ising import ising

#data implementation
# _____
—

#np.loadmat(Culture1DIV6, mdict=None, appendmat=True, **kwargs)
mat_contents = io.loadmat('Culture5DIV14.mat')
#print(mat_contents)
#a=np.histogram(mat_contents,bins='auto')

#data filter
# _____
—
spike_data=mat_contents['data']['spikes'][0][0]

spikes=[]
Nspikes=[]
Tmax=0
for s in spike_data:
    spikes+=s[0][0]
    Nspikes+=len(s[0][0])
    if np.max(s[0][0])>Tmax:
        Tmax=np.max(s[0][0])

dT=0.00005
Tmax=3600/dT
print(Tmax)
```

#information analysis

#

---

w=20E-3

r=w/dT

T=int(Tmax/r)

N=len(spikes)

X=np.ones((N,T))\*(-1)

order=np.argsort(Nspikes[::-1])

f=len(X)

for n,i in enumerate(order[0:N]):

    for s in spikes[i]:

        ind= int(np.floor(s/r))

        X[n,ind]=1

plt.figure()

plt.imshow(X, aspect='auto',interpolation='nearest')

plt.xlabel('spikes')

plt.ylabel('spikes length ')

plt.show()

size=N

m=np.zeros(size)

c=np.zeros((size,size))

C=np.zeros((size,size))

m=np.mean(X,axis=1)

for i in range(size):

    for j in range(i+1,size):

        c[i,j]=np.mean(X[i,:]\*X[j,:])

        C[i,j]=c[i,j]-m[i]\*m[j]

l2=0

for i in range(size):

    for j in range(i+1,size):

        l1=C[i,j]

        if (l1>l2): l2=l1

for i in range(size):

    for j in range(i+1,size):

        C[i,j]=C[i,j]/l2

q=20

k=q+1

D=np.zeros(k)

```

sum=0

for i in range(size):
    for j in range(i+1,size):
        if (C[i,j] != 0):
            # d=math.fabs((C[i,j]*k)//1)
            d=abs(int((C[i,j])*q))
            # print(C[i,j],d)
            D[d]=D[d]+1
            sum+=1

D=D/sum
C=C*I
print(D)
y_pos=np.linspace(0,0.9,k)
plt.imshow(C, aspect='auto',interpolation='nearest')
plt.colorbar()
plt.title('activity matrix')
plt.show()
index=np.arange(k)
width=0.01
opacity=1
probability = plt.bar(y_pos, D, width, alpha=opacity, color='r', label='Probability of the different
states')

plt.xlabel('activity')
plt.ylabel('Probability')

```

#graphs and error

#

---

```

def graph(formula, x_range):
    x = np.array(x_range)
    y = eval(formula)
    plt.plot(x, y)
#     plt.show()

```

X=0

Y=0

Z=0

```

def funcritic(x, a, b, c):
    global X,Y,Z
    X=a
    Y=b
    Z=c

```

```

    return a*1/(x**(-b))+c
popt, pcov = curve_fit(func, y_pos, D)
graph('X*1/(X**(-Y))+Z', np.linspace(0, 1, num=k**2))
green_patch = mpatches.Patch(color='green', label='potential')

```

```

A=0
B=0
C=0
def func(x, a, b, c):
    global A,B,C
    A=a
    B=b
    C=c
    return a * np.exp(-b * x) + c

```

```

popt, pcov = curve_fit(func, y_pos, D)
print(A,B,C)
graph('A* np.exp(-B* x) + C', np.linspace(0, 1, num=k**2))
blue_patch = mpatches.Patch(color='blue', label='exponential')
plt.legend(handles=[blue_patch, green_patch])
plt.show()

```

```

l=len(D)
error=np.zeros(l)
for i in range(len(D)):
    error[i]=np.absolute((X*1/(i**(-Y))+Z)-D[i])

criticerror=np.zeros(l)
for i in range(len(D)):
    criticerror[i]=np.absolute((A* np.exp(-B* i) + C)-D[i])

```

```

plt.plot(y_pos, error, 'o')
plt.plot(y_pos, criticerror, 'x')
plt.legend(handles=[blue_patch, green_patch])
plt.xlabel('activity')
plt.ylabel('error')
plt.show()

```

